sinclair

ZX Spectrum+

מדריך למשתמש



תוכנת ספקטרום

כל מגוון התוכנה עבור מחשבי ספקטרום (כולל כל הנושאים הקיימים כיום) תואם בשלמותו למחשב החדש ספקטרום +ZX.

גאים להציג את הספקטרום +ZX

חברת סינקלייר ריסרץ' המובילה בטכנולוגיית המחשב הזעיר הביאה את המחשב לכל בית. בעקבות פיתוחו של המחשב הזעיר הראשון בעולם שהיה שווה לכל כיס, דגם ZX80, הושגו יכולת חישוב גדולה עוד יותר ותכונות משופרות נוספות – במחשבי ZX81, ספקטרום ZX ו־QL. מילת המפתח היתה תמיד הנוחות בשימוש, גם בתכנון המוצרים וגם בדרך פעולתם.

הספקטרום +ZX מצעיד את סינקלייר ריסרץ' צעד נוסף בדרך זאת. בתוכו, תוכל למצוא מכונה המשלבת את כל מעלות הספקטרום בגירסה משופרת ההופכת מחשב פופולרי ורב־עצמה זה לנוח בשימוש עוד יותר מקודמו. תקוותנו היא שתוכל לנצל את מלוא האפשרויות שמעניק המחשב החדש שלך

Clive Sinclair

תוכן העניינים

נכתב ע"י ניל ארדלי
הוצא לאור ע"י דורלינג קינדרסלי בע"מ
בשיתוף עם
סינקלייר ריסרץ' בע"מ

כיצד להשתמש בספר זה

מדריך זה עבור מחשב ספקטרום +ZX מכיל 4 פרקים בקידוד צבעוני. כדי לפנות לפרק מסויים, פשוט פתח את הספר בחלק עם הצבע הנכון.

1 ההתחלה

הכנת הספקטרום +ZX לפעולה ■ כוונון מקלט הטלביזיה ■ איתור תקלות הכנה לפעולה ■ מה מסוגל הספקטרום +ZX לעשות ■ כיצד להשתמש בתכנה מוכנה לריצה ■ כיצד לטעון תכנית ■ איתור תקלות טעינת תכנה.

2 צעדים ראשונים בתכנות

המקלדת – לוח הבקרה של המחשב ■ כיצד להפעיל את המקשים ■ מחשבון הטלביזיה ■ הצבע וכיצד להשתמש בו ■ גרפיקת 'עשה זאת בעצמך' ■ לוח טיוטה על המסך ■ תכנן בעצמך צורות ותמונות שונות ■ כיצד ליצור תווים משלך במחשב ■ הנפשה ■ כיצד ליצור מוסיקה ואפקטים קוליים ■ כיצד לשמור את תכניותיך ■ מאתר תקלות איחסון תכנה.

3 למד על הספקטרום +ZX

מה נמצא בפנים? ■ כיצד פועל הספקטרום +ZX ? ■ כיצד לחבר רכיבים היקפיים ■ מיפוי זכרון בספקטרום +ZX.

4 למד על הבייסיק של סינקלייר

מדריך עיון למילות מפתח עבור מתכנת בייסיק של סינקלייר ■ דוחות מסך בספקטרום ■ מה מעבר לביסיק ■ מילון מונחי מחשב

ההתחלה

פרק זה יורה לך כיצד להתחיל ולגלות את יכולתו של הספקטרום +ZX. כאן תמצא כיצד להכין את המחשב לפעולה כך שיגיב מיידית על דרישותיך, ואז בידך הברירה. אתה יכול להקיש פנימה מספר תכניות אשר יצעידו את הספקטרום דרך מרבית פעולותיו ויציגו את מלוא יכולתו בגרפיקה ואפקטים קוליים, או שתוכל ללמוד כיצד להשתמש בתכנה מוכנה לריצה, כגון משחקי מחשב. כך או כך, אתה תיהנה במהרה מהמחשב החדש שברשותך.

הכנת הספקטרום ZX+ לפעולה

כדי להכין את הספקטרום לפעולה, עבור תחילה על רשימת התיוג שלהלן כדי להבטיח שיש ברשותך את כל החלקים הדרושים. אחר כך פעל לפי ההוראות בעמוד שממול לחבורים ולהתחברות לרשת החשמל. בשלב זה אינך זקוק לרשמקול.

דאג שכל החיבורים יהיו יציבים. אם במקרה תנתק את החשמל בזמן שהספקטרום פועל, תלך התכנית לאיבוד וכן כל המידע או התוצאות.

לאחר גמר השימוש במחשב, נתק ליד השקע בקיר אם יש עליו מפסק, והוצא את התקע מהשקע.

רשימת תיוג: האם יש ברשותך כל מה שנדרש?



1 מחשב ספקטרום ZX+

2 ספק כח ZX — מייצר אספקת 9 וולט ז"י הנדרש ע"י הספקטרום.

3 כבל אנטנה — מחבר את הספקטרום למקלט הטלביזיה.

4 כבל קלטת — מחבר את הספקטרום לרשמקול.

5 תעודת אחריות — עליך למלא אותה ולהחזירה לסוכן או ליצרן.

6 קלטת המכילה מדריך למשתמש.

7 ספר הדרכה זה.

מה שאתה נדרש לספק:

1 מקלט טלביזיה

2 רשמקול

3 תקע לרשת החשמל הביתית.

שאלות ותשובות לשלב ההכנה לפעולה

האם אני חייב להשתמש במקלט טלביזיה צבעוני?
לא. ברם, לא תוכל לראות את הצבעים שמפיק הספקטרום במכשיר שחור־לבן.

האם כל מקלט טלביזיה מתאים?
הספקטרום אמור לתת תמונה עם כל מקלט טלביזיה שברשותך. במידה ולא, יתכן שמקלט הטלביזיה והמחשב משתמשים במערכות צבע שונות. הדבר עלול לקרות אם מקלט הטלביזיה ישן מאד, או אם המקלט והמחשב נרכשו במדינות שונות. אם אתה בספק, פנה ליצרן מקלט הטלביזיה.

האם אני יכול להשתמש בצג במקום מקלט טלביזיה?
כן. ניתן להשיג צגים המפיקים תמונה באיכות מעולה עם הספקטרום.

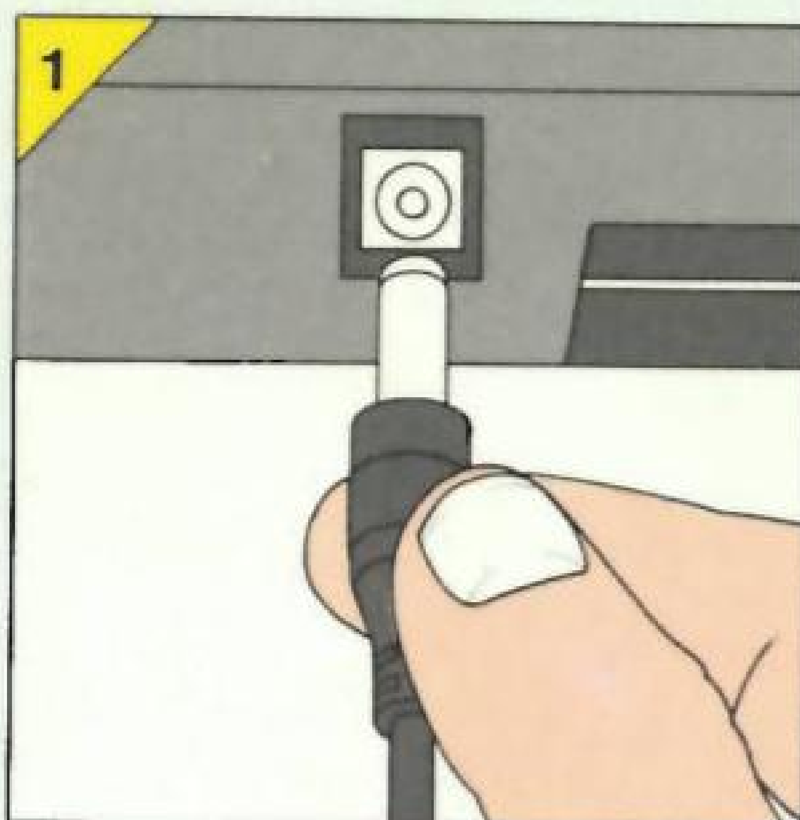
איזה תקע דרוש לרשת החשמל?
הספקטרום צורך 1.4A במתח של 240V/50HZ, מתח הרשת התיקני בבריטניה.

האם הספקטרום מייצר הפרעות?
הספקטרום עלול להפריע לרדיו הנמצא בקרבת המחשב. לא ייגרם כל נזק לרדיו או למחשב.

האם אני יכול להשתמש בראם ZX 16K?
לא. מחסנית ראם זו מיועדת לשימוש במחשב סינקלייר ZX81 בלבד.

הפעלת הספקטרום +ZX

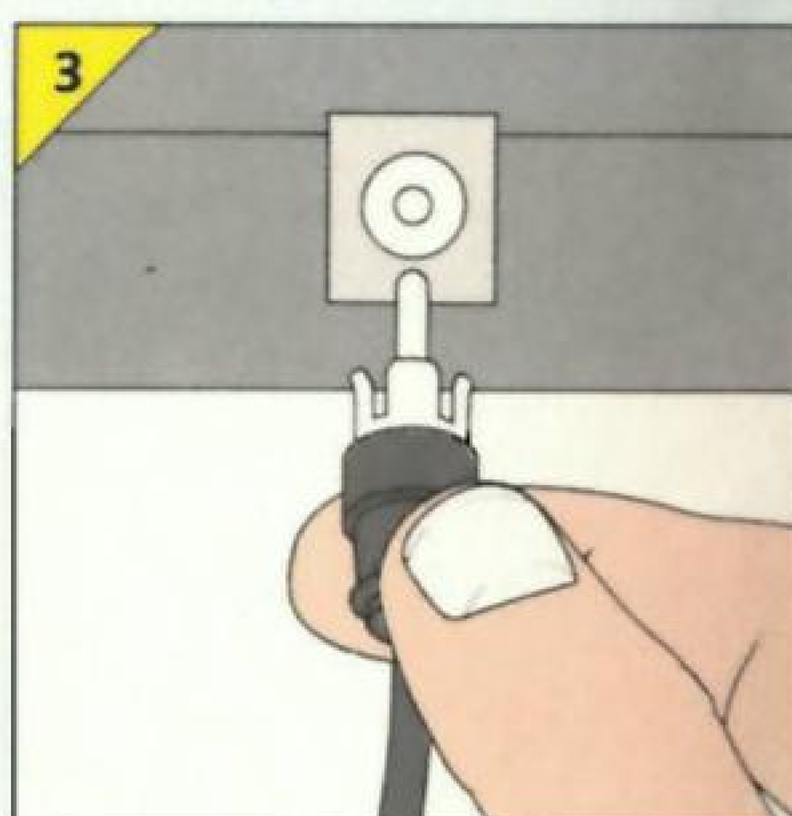
התחל בהתאמת תקע חשמלי לשני הקצוות החשופים של החוטים מספק הכח. כמו כן עליך להתאים נתיך 3A עבור התקע. שים לב לכך שהספקטרום אינו דורש חיבור הארקה לאדמה על אף שיתכן ולתקע שברשותך יש 3 פינים. אחר כך עקוב אחר סדר הפעולות המאוייר לחיבור הספקטרום לרשת החשמל הביתית ולמקלט הטלביזיה שלך. עם השלמת כל החיבורים והפעלת המערכת, עבור לעמוד הבא כדי ללמוד על כוונון מקלט הטלביזיה.



הכנס את התקע לשקע רשת החשמל והפעל אותו אם יש עליו מפסק. לספקטרום אין מפסק הפעלה משלו.



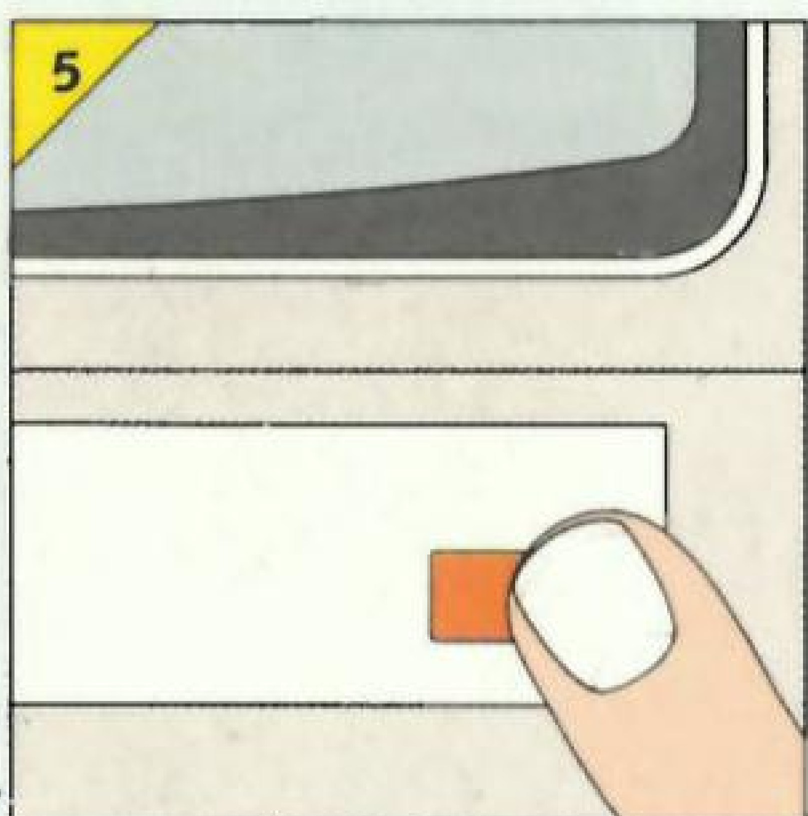
הכנס את התקע הקטן שעל כבל ספק הכח לשקע המסומן 9VDC בספקטרום



הכנס את כבל האנטנה לשקע המסומן TV בספקטרום. רק אחד מהתקעים בכבל האנטנה יתאים לשקע זה.



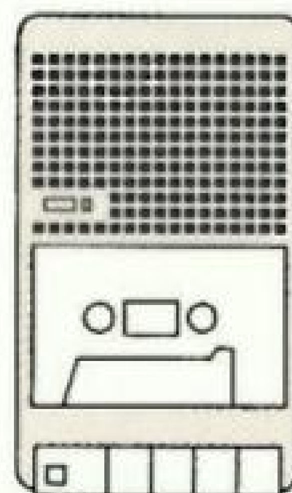
נתק את כבל האנטנה ממקלט הטלביזיה שלך. הכנס את התקע השני בכבל האנטנה של הספקטרום לשקע האנטנה במקלט הטלביזיה.



הפעל את מקלט הטלביזיה וסובב מייד את פקד עוצמת הקול כלפי מטה. אתה מוכן עתה לשלב כוונון המקלט כך שיקלוט את אותות הספקטרום.

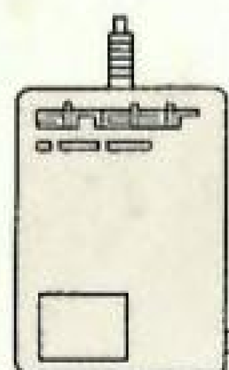
שקעים ומחברים בספקטרום

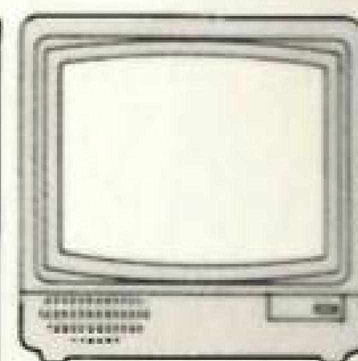
שקע אזנייה (EAR) — שקע האזנייה של רשמקול קלטות מתחבר לכאן במטרה לטעון בספקטרום תכניות שהוקלטו על קלטות.



מחבר קצה — ניתן לחבר כאן מגוון רחב של חמרה, כולל כונן זעיר מדפסת, ומודם.

שקע כח (POWER) — אספקת 9 וולט ד"ז המיוצרת ע"י ספק הכח ZX. השנאי מחובר דרך שקע זה.





שקע טלביזיה (TV) — כאן מתחבר שקע האנטנה של מקלט טלביזיה כך שניתן יהיה לראות את תמונות הספקטרום על המסך.

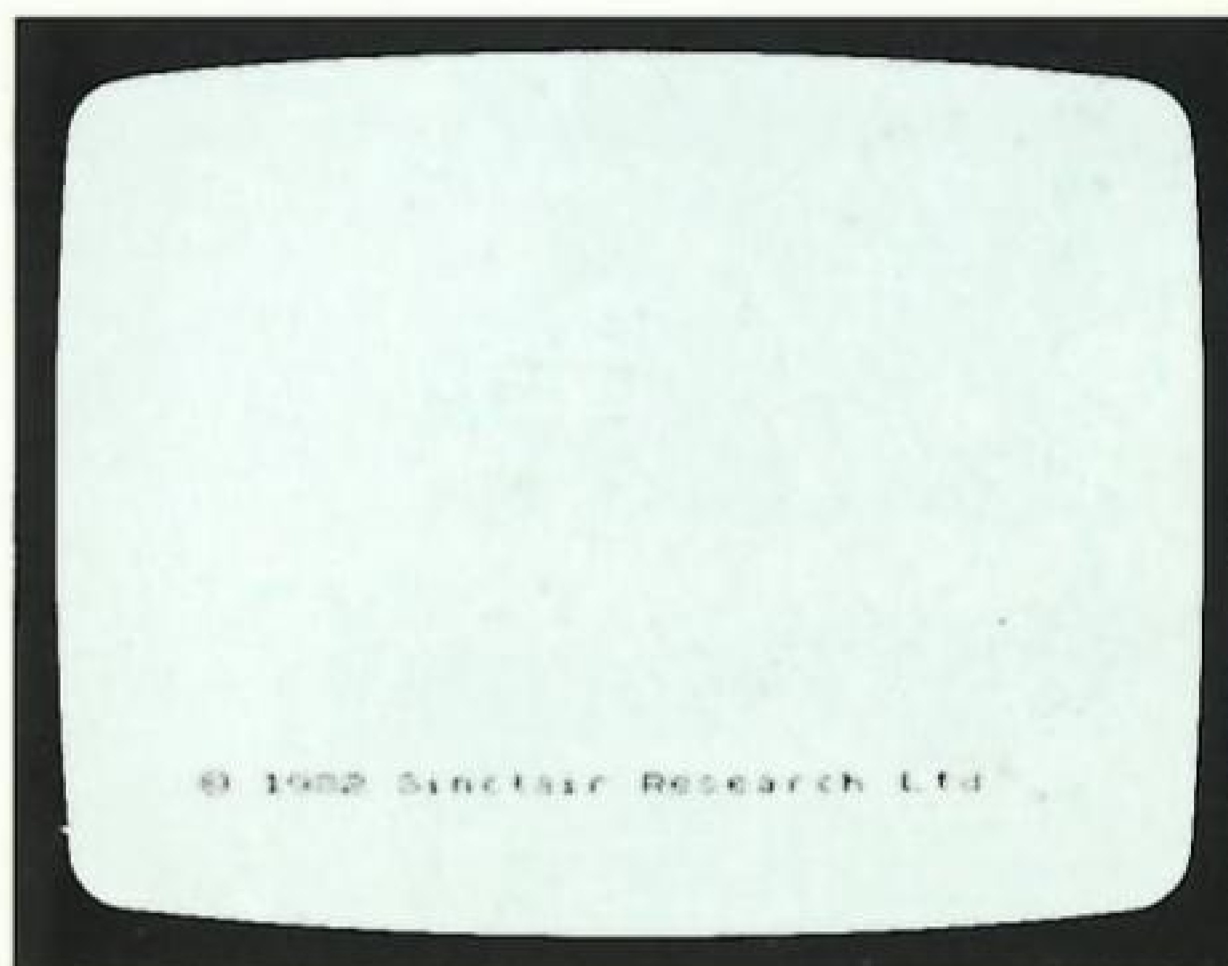
שקע מיקרופון (MIC) — כאן מתחבר שקע המקרופון של רשמקול קלטות במטרה לשמור (להקליט) תוכניות על קלטות.



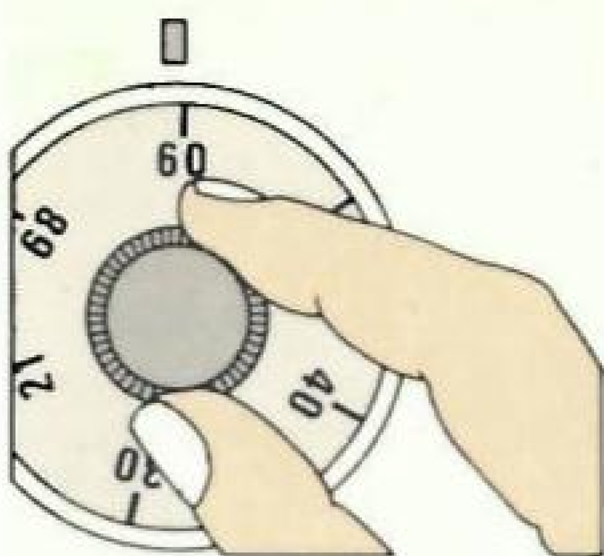
כוונון מקלט הטלביזיה

מחשב הספקטרום מפיק אות ווידאו לטלביזיה צבעונית בתדר המתאים לערוץ 36 בתחום התדרים UHF, כך שמקלט הטלביזיה חייב להיות מכוון לערוץ זה כדי לקלוט את תמונות המחשב.

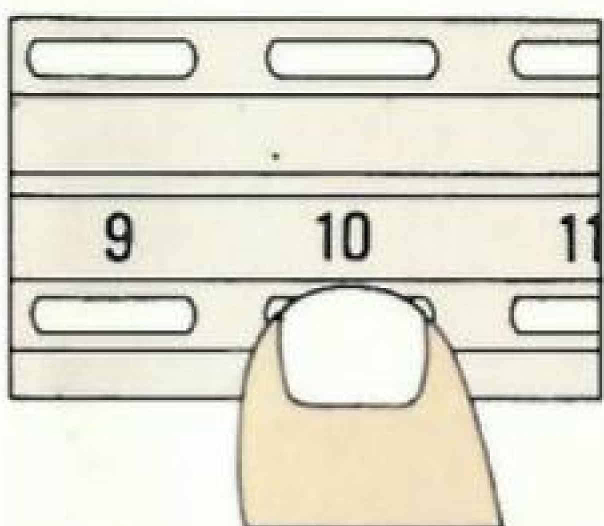
לאחר הפעלת הספקטרום וחיבורו למקלט הטלביזיה, כוונן את פקד הכוונון במקלט עד שתקבל את הודעת הספקטרום כמתואר במסך הראשון להלן. לאחר שתראה זאת, אתה מוכן לבדיקת צבעי הספקטרום והתחלת התכנות. אם אינך מקבל את הודעת 'זכויות יוצרים', או אם הצבעים אינם נראים באיכות טובה, עיין בטבלת איתור תקלות בעמוד הבא.



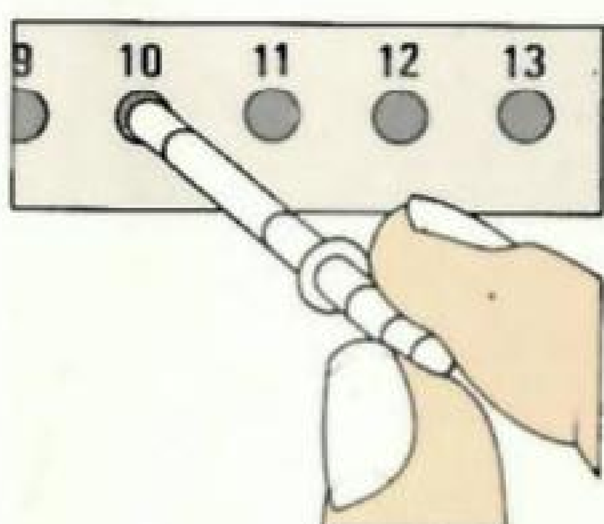
פקדי כוונון

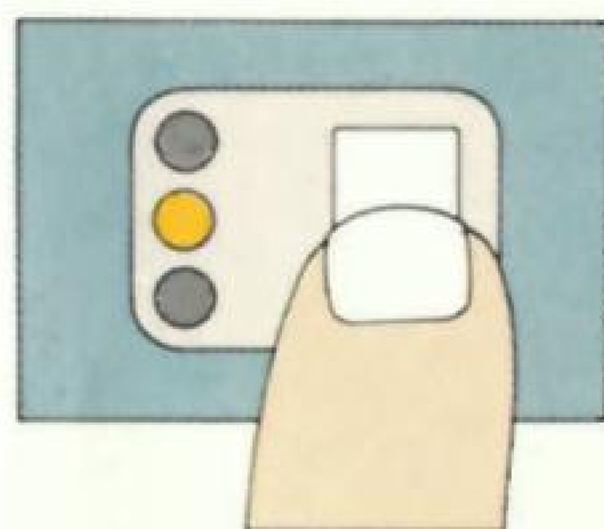


כוונון משתנה — פקד כוונון משתנה בורר כל ערוץ. פשוט סובב את הכפתור עד לקבלת הודעת הספקטרום.



כוונון בלחצנים — בחר כפתור כוונון עבור המחשב וכוונן אותו עד לקבלת הודעת המחשב. במידת האפשר, השתמש בכפתור פנוי, כך לא תצטרך לכוונן את המקלט בכל פעם שהמחשב מופעל.





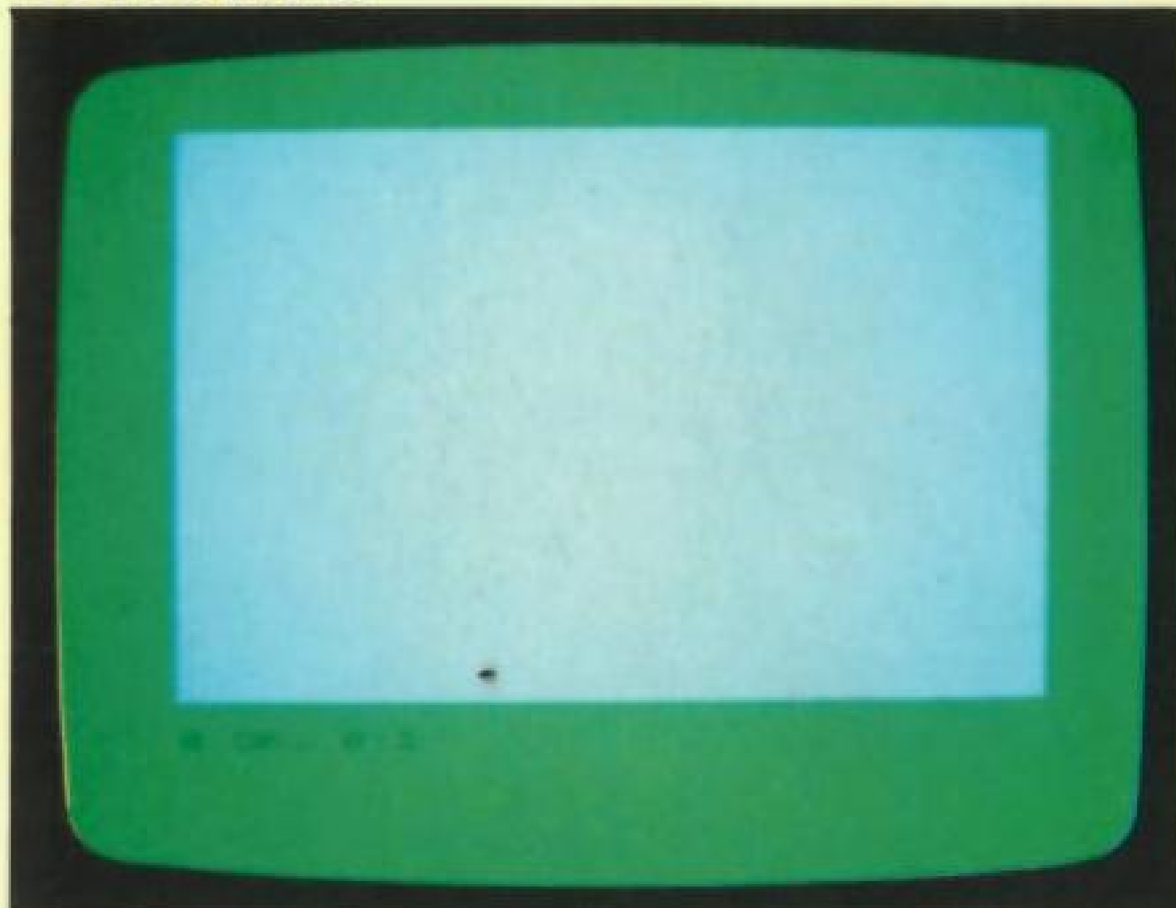
כוונון אלקטרוני — במערכת זאת, המקלט עצמו מתכוונן לערוץ הנדרש. מקלטי טלביזיה עם ברירת ערוצים סינתטית ללא אפשרות כוונון ידנית עלולים להיות לא מתאימים לשימוש עם מחשב זה.

כיצד לבדוק את צבעי הספקטרום

כדי לבדוק את צבעי הספקטרום, פשוט לחץ על מקש B ואחר כך על מספר מ־1 עד 6. הודעת 'זכויות יוצרים' תיעלם; ראשית תופיע המילה BORDER ואחריה המספר. כעת לחץ על מקש ENTER, איזור ה'גבול' שבמסך ישתנה לצבע המסומן במקש המספר. המסכים שלהלן מתארים מה קורה

לאחר הקשת BORDER 4 ו־ENTER, ואח"כ BORDER 3 ו־ENTER. הקשת BORDER 7 תחזיר את איזור הגבול לרקע לבן.

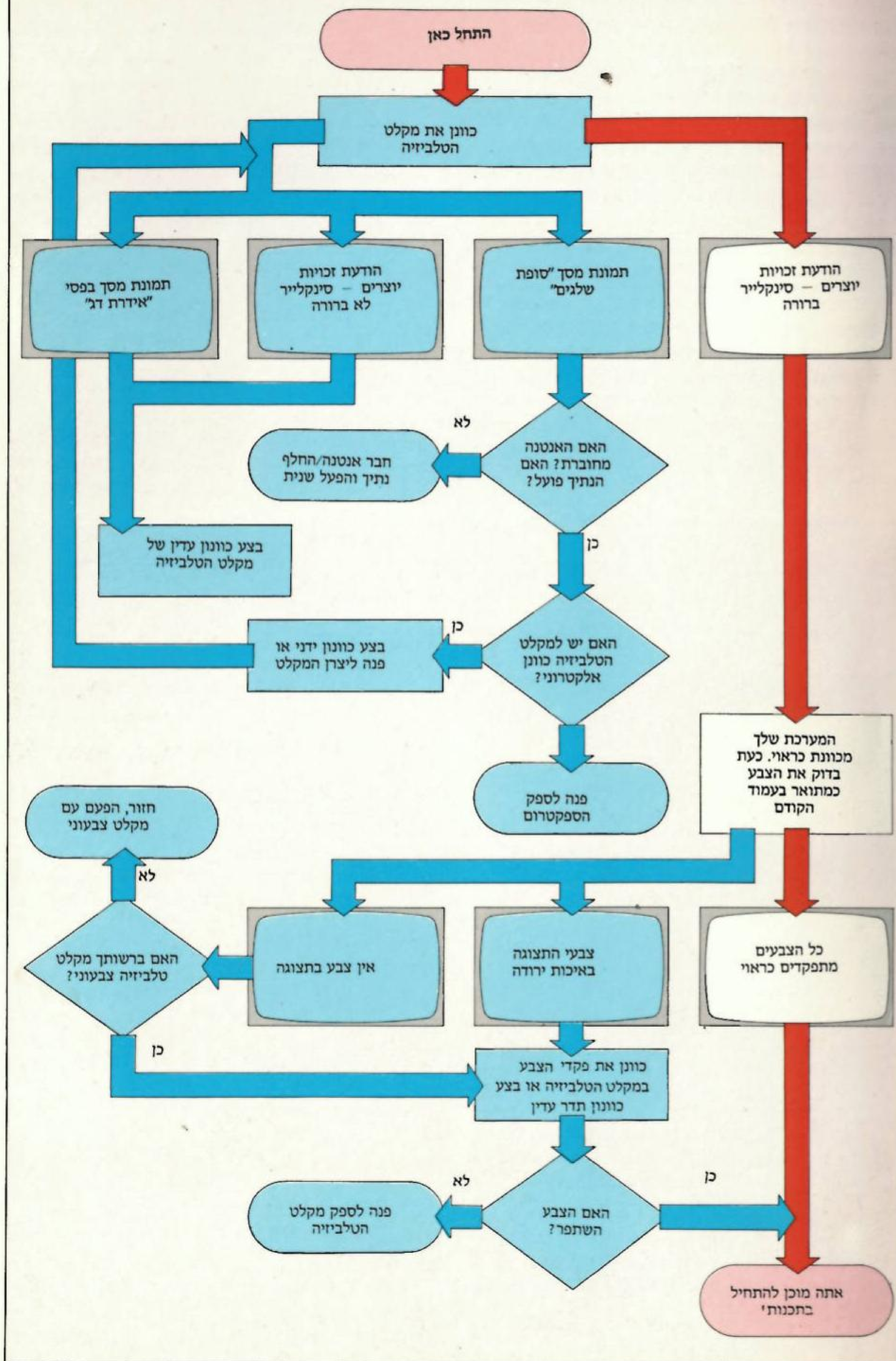
BORDER 3



BORDER 4



איתור תקלות בשלב הכנה לפעולה



מה מסוגל הספקטרום +ZX לעשות?

ראשית, נסיון

כעת, לאחר שהפעלת את הספקטרום ומקלט הטלביזיה מכוונן כראוי, נסה ללחוץ על מספר מקשים. על המסך יופיעו מילים ואותיות, ויתכן שגם מספרים.

ברם, אם אינך יודע כיצד לתכנת את הספקטרום, קרוב לוודאי שהמחשב כלל לא יגיב להקשותיך. אך אל דאגה – שום נזק לא יכול להגרם, ולא חשוב על אילו מקשים לחצת.

כעת לחץ על מקש RESET בצד שמאל של המחשב, ואתה מוכן להפעלת הספקטרום. ארבעת העמודים הבאים יציגו בפניך מספר הדגמות על מסך הטלביזיה, שיראו מה הספקטרום מסוגל לעשות.

כיצד להקיש

כדי להקיש מילה כלשהי, אות או מספר, שים לב תחילה למצבה על המקש. פעל לפי סדר הפעלת מקשי הברירה המתואר להלן

מילת מפתח תחתונה או סימן – לחץ EXTEND MODE, החזק מקש SYMBOL SHIFT לחוץ, ולחץ על המקש הדרוש.

מילת מפתח עליונה – לחץ EXTEND MODE ואח"כ על המקש.



מילת מפתח עליונה (חלק מורם) – לחץ על המקש.

מילת מפתח תחתונה או סימן (חלק מורם) – החזק מקש SYMBOL SHIFT לחוץ ולחץ על המקש הנדרש.

אות או מספר (חלק מורם) – לחץ על המקש הנדרש. החזק מקש CAPS SHIFT לקבל אותיות גדולות.

ניתן למצוא פרטים מלאים על הפעלת המקשים בעמ' 20-21.

כעת, תכנת את הספקטרום

הספקטרום מסוגל לבצע דברים רבים. אך כדי שהוא יפעל, עליך לספק לו מערך פקודות הנקרא תכנית מחשב.

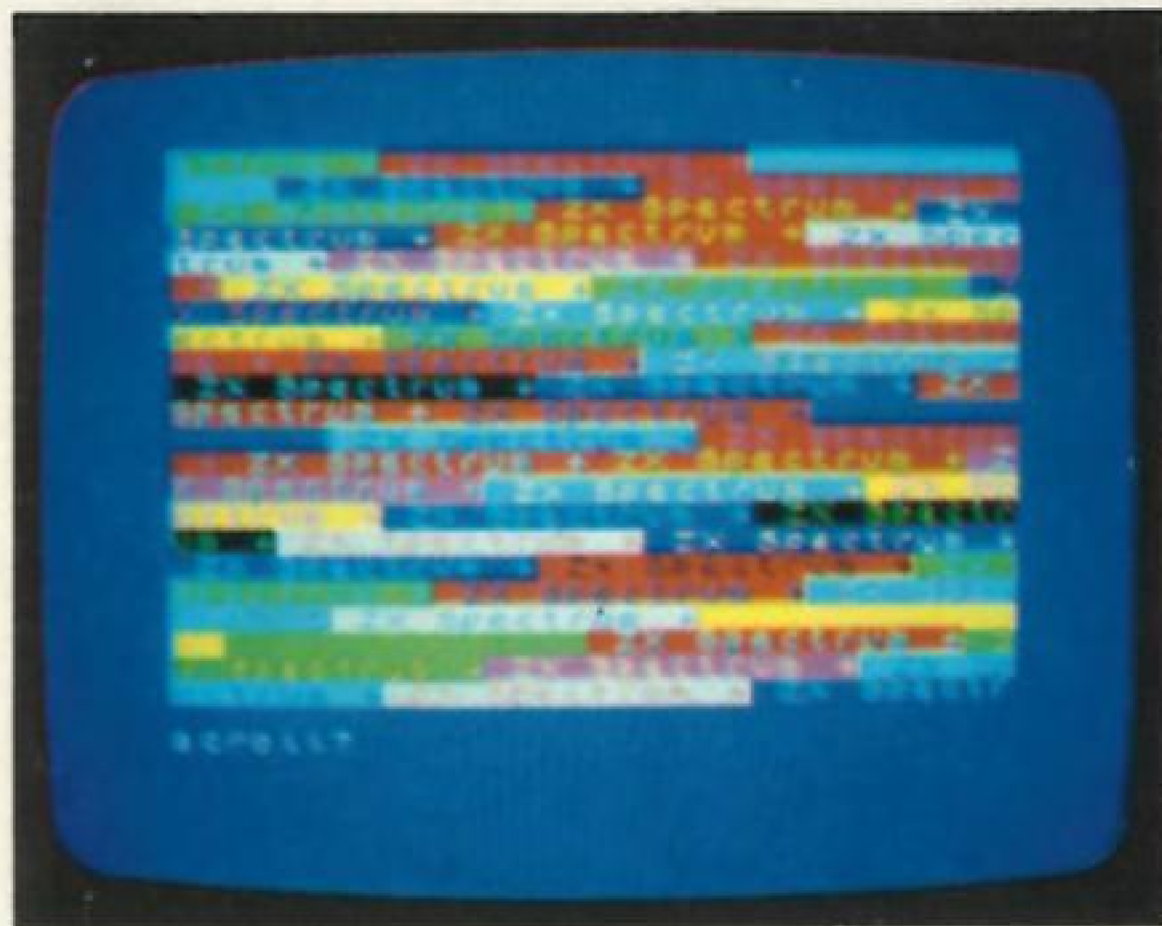
להלן מופיע אוסף תכניות קצרות שיצעידו את הספקטרום דרך שלבי פעולה שונים. כל שעליך לעשות הוא להקיש את פקודות התכנית **בדיוק** כפי שהן מופיעות כאן. תמונות המסכים מתארות למה עליך לצפות, אך אם תקרא את הפיסקה 'כיצד לשנות תכנית' בעמוד הבא, תוכל לבצע בעצמך ניסויים בתכניות.

כיצד להקיש ולהריץ תכנית

כל מערך פקודות מתואר ברשימה הנקראת רישום (LISTING). אתה תראה שרישומי התכניות מכילים מספר חלקים שכל אחד מהם מתחיל במספר – 10, 20, וכן הלאה. כל חלק נקרא שורה בתכנית (אפילו אם הוא תופס שתי שורות במסך), ומכיל פקודה אחת או יותר למחשב. בכל שורת תכנית, תראה מילים שלמות או קיצורים המכילים שתי אותיות או יותר, כגון PRINT, LET, RND, PI, PAPER ו־GO TO.

שמות

```
10 BORDER 1: INK RND*7
20 PAPER RND*7
30 PRINT " ZX Spectrum +";
40 GO TO 10
```



השם ספקטרום +ZX מופיע בצבעים רבים על פני כל המסך. המחשב עוצר ומופיעה הודעת ?SCROLL בתחתית המסך. כדי לאפשר לתצוגה "גלגול" כלפי מעלה, לחץ על מקש כלשהו מלבד N, SPACE, BREAK או STOP. אם אתה עוצר את הגלגול, ולוחץ BREAK, ואח"כ R(RUN) מלווה ב־ENTER. השמות יופיעו בצורות צבעים שונות.

נסה זאת

בשורה 30, שנה "Zx Spectrum+" לשמך במרכאות (") – למשל

30 PRINT"John ";

זכור לכלול נקודה ופסיק (;). שמך יופיע מפוזר על גבי כל המסך.

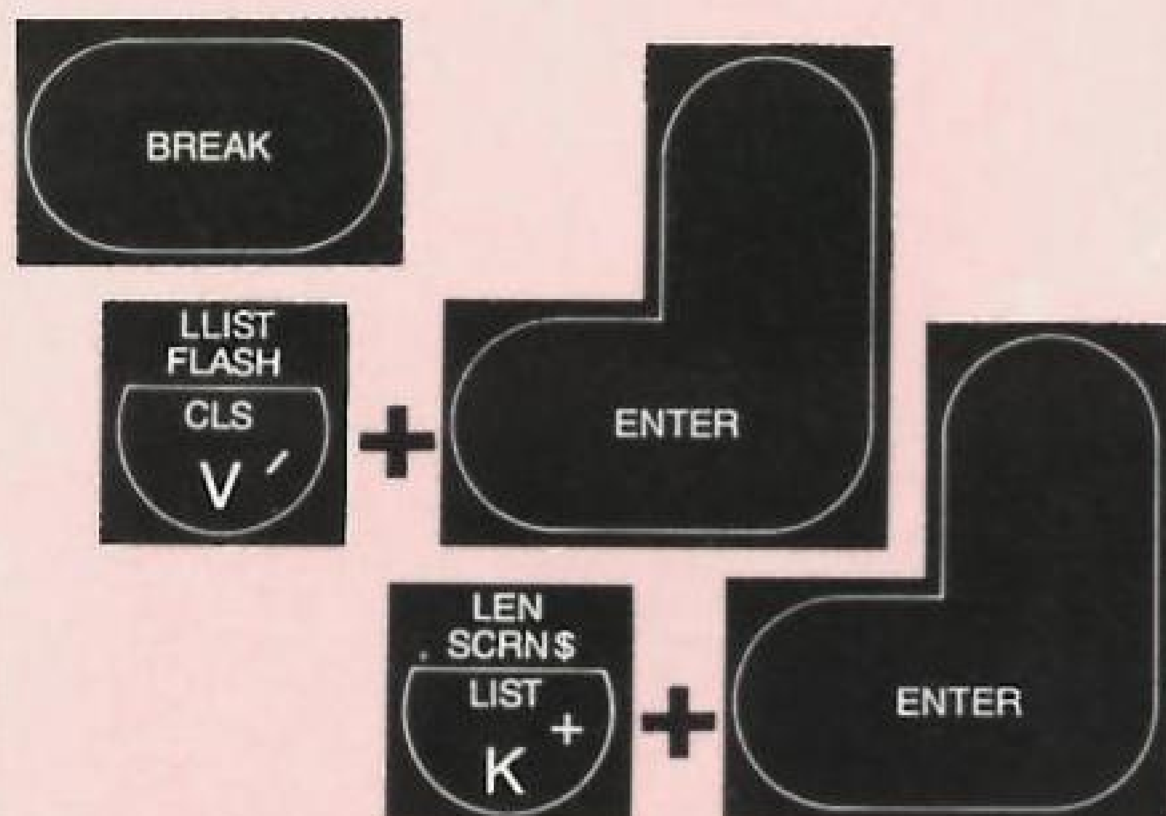
אלה נקראות מילות מפתח ואינך חייב להקישן אות אחרי אות. במקום זאת, מצא את המקש במקלדת אשר נושא את מילת המפתח (לדוגמא, PRINT נמצאת על מקש P), ומלא אחר ההוראות המפורטות בפיסקה 'כיצד להקיש'.

כאשר אתה מקיש שורה, היא מופיעה בתחתית המסך. כאשר אתה מגיע לסוף שורת התכנית, לחץ על מקש ENTER. השורה תופיע כעת בראש המסך. כעת הקש והכנס כל שורה באותה צורה. אם תקיש בטעות מקש לא נכון, עיין בפיסקה 'כיצד לתקן טעויות' בעמוד הבא.

כאשר סיימת להקיש את כל השורות, לחץ R. מילת המפתח RUN תופיע. כעת לחץ ENTER והספקטרום יזנק לפעולה עם הרצת התכנית.

כיצד לשנות תכנית

המתן עד לגמר התכנית או הקש BREAK. אח"כ לחץ V(CLS) ואז ENTER מלווה ב־K(LIST) ו־ENTER. רישום התכנית (רשימת השורות) תופיע על המסך.



זהה איזו שורה אתה רוצה לשנות, ואז הקש את כל השורה החדשה, כולל מספר השורה, ולחץ ENTER. השורה החדשה תחליף את מקום הישנה. לחץ R(RUN) ו־ENTER, והתכנית החדשה תרוץ.

צורות

```
10 LET a$=""
20 FOR x=1 TO 7
30 LET a$=a$+CHR$ (RND*14+129)
40 NEXT x
50 INK RND*7
60 BORDER RND*7
70 PRINT a$;
80 GO TO 70
```



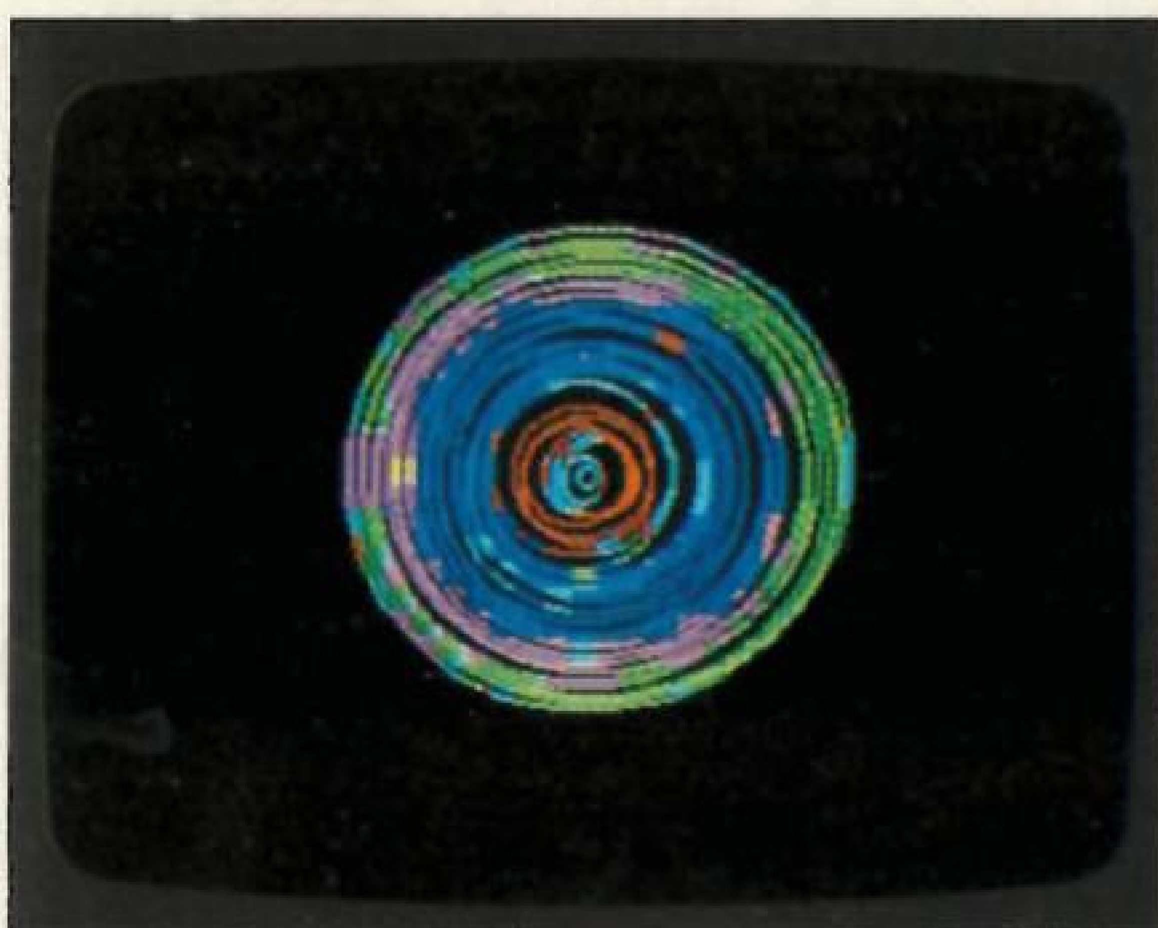
עם הרצת התכנית, תופיע על המסך צורה גיאומטרית צבעונית. כאשר המסך מלא, התצוגה תעצור בהודעת ?SCROLL. כדי לראות יותר מתצוגה זאת, הקש משהו (מלבד N, SPACE, BREAK או STOP) כדי לגרום לצורה לעלות כלפי מעלה. כדי לראות צורה חדשה בצירוף צבעים אחר, הקש N עם הופעת הודעת ?SCROLL. אח"כ הקש BREAK, מלווה ב־R(RUN) וב־ENTER.

נסה זאת

בשורה 20, שנה 7 למספר אחר כדי לקבל צורה אחרת. נסה 8, לדוגמה.

עיגולים מבזיקים

```
 10 BORDER 0: PAPER 0: CLS
 20 CIRCLE INK RND*6; FLASH RND
;120+RND*8,80+RND*6,RND*80
 30 BEEP 0.1,RND*60
 40 IF RND>.9 THEN GO TO 60
 50 GO TO 20
 60 FOR y=-2 TO 4
 70 FOR x=0 TO 6
 80 BORDER x
 90 BEEP .05,x*y
100 NEXT x
110 NEXT y
120 RUN
```



על המסך נבנה מערך של עיגולים מבזיקים כמעט קונצנטריים במגוון צבעים שונים. ואז, בבת אחת הגבול מבזיק, המחשב מפיק צליל שריקה, ומערך עיגולים חדש מופיע.

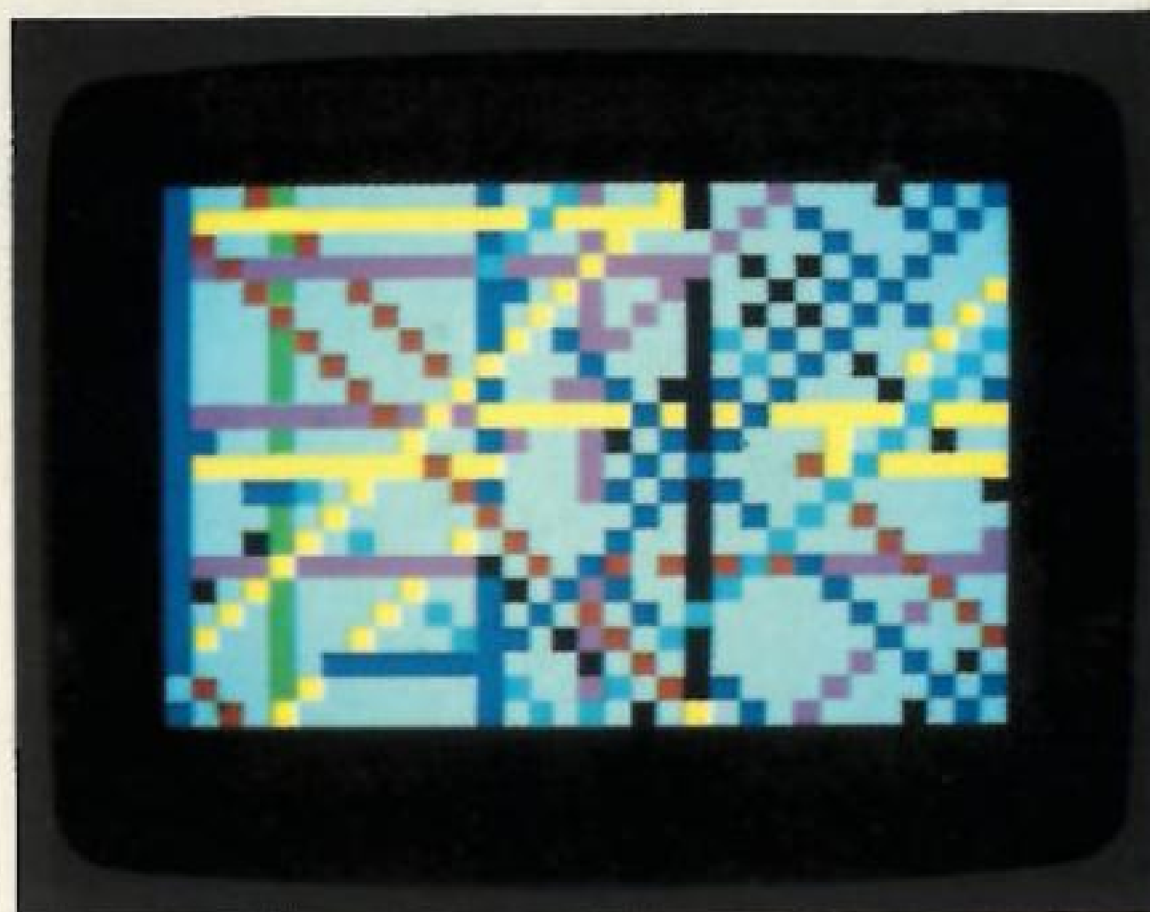
נסה זאת

לפני רישום התכנית (בעזרת מקש K), הקש פנימה PAPER 7 ולחץ ENTER.

אח"כ הקש שנית את שורה 20, הפעם בהשמטת שתי מילות המפתח FLASH RND; ואז העיגולים לא יבזיקו עוד.

מוזאיקה מטורפת

```
  5 BORDER 0: CLS
 10 LET h=16: LET v=11
 20 LET x=INT (RND*3-1): LET y=
INT (RND*3-1)
 30 INK RND*7
 40 FOR z=1 TO 20
 50 PRINT AT v,h;CHR$ 143
 60 LET h=h+x
 70 LET v=v+y
 80 IF h<0 THEN LET h=31
 90 IF h>31 THEN LET h=0
100 IF v<0 THEN LET v=21
110 IF v>21 THEN LET v=0
120 NEXT z
130 GO TO 20
```



ריבוע צבעוני מתרוצץ הלוך ושוב על פני כל המסך, ובונה צורה צבעונית. בכל פעם שתפעיל מחדש את התכנית, תיבנה צורה חדשה.

נסה זאת

בשורה 50, שנה 143 ל־42 ואז תראה כוכבים! נסה מספרים אחרים בתחום 33 עד 142. עיין בטבלת מערך התווים לבדיקת התווים הצפויים.

כיצד להתחיל מחדש תכנית

כמה מתכניות אלה – כמו 'פסים וכוכבים' – מסתיימות ומפיקות את הדוח 0 OK, עם מספר השורה האחרונה בתכנית. הדוח מציין שכל התכנית הסתיימה. כדי להתחיל מחדש, פשוט לחץ R(RUN) ו־ENTER.

תכניות אחרות ממשיכות לרוץ, כמו 'מוזאיקה מטורפת', או מתחילות שנית באופן אוטומטי, כמו 'זריחה מנצנצת'.







כדי לעצור תכניות כאלה, לחץ BREAK.
המשך ללחוץ על מקש זה עד שהתכנית נעצרת ומופיע דיווח BREAK על מנת להתחיל מחדש, פשוט לחץ R(RUN) ו־ENTER.

כיצד לתקן טעויות

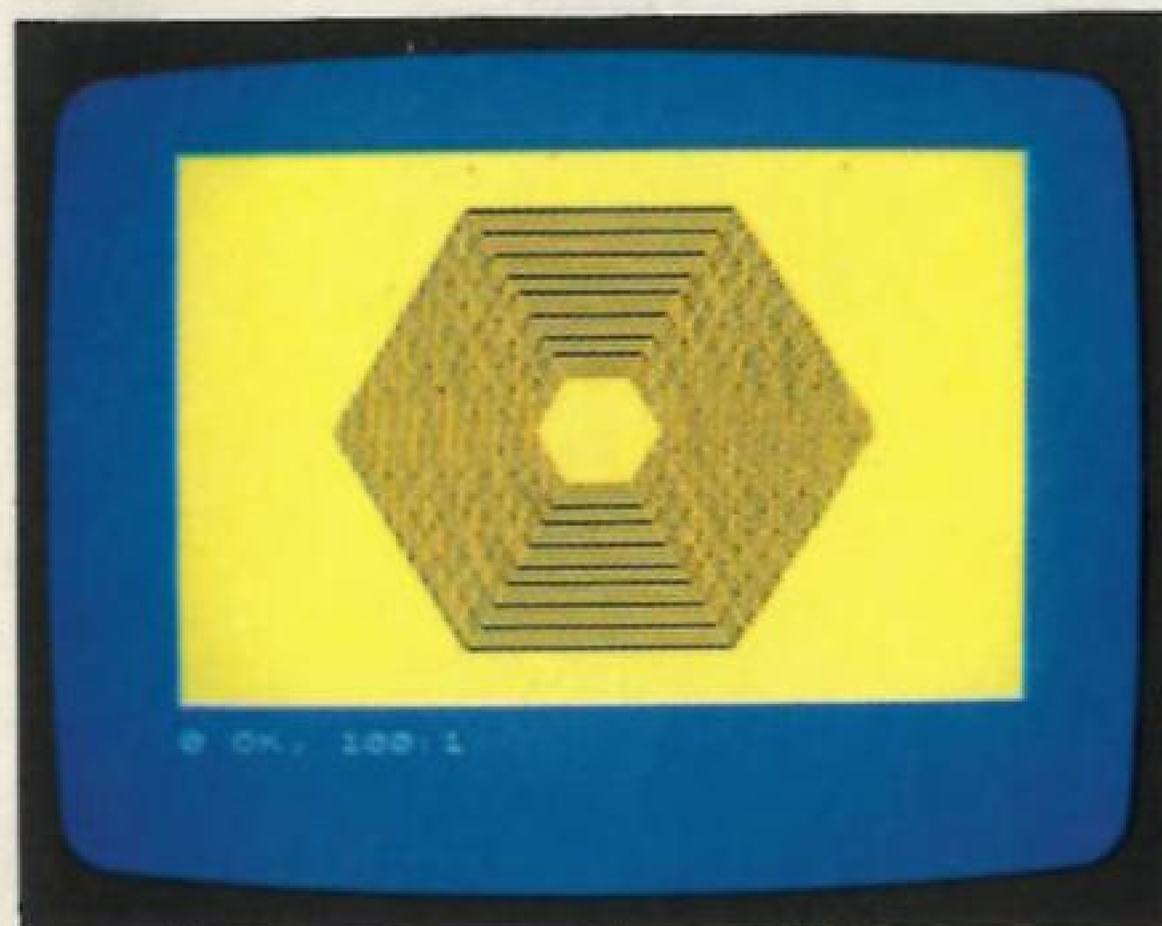
אם אתה לוחץ על מקש לא נכון או לא לוחץ כנדרש על מקשי SHIFT או EXTEND MODE, אל תדאג. לחץ על מקש DELETE, ומילת המפתח, סימן, אות או מספר, האחרון שהוקש יעלם. המשך ללחוץ על DELETE כדי למחוק תווים נוספים.



אם בצעת טעות לאחר לחיצת ENTER, עלול להופיע סימן שאלה מהבהב לפני השגיאה. לחץ DELETE למחיקת השורה עד מקום השגיאה וכולל השגיאה, ואז השלם את השורה כראוי ולחץ ENTER. אם הצלחת להכניס שורה לא נכונה, התכנית עלולה להפסיק לפעול, ולהפיק דוח בתחתית המסך המציין את מספר השורה השגוייה. הקש את השורה המתוקנת במלואה, ולחץ ENTER, R(RUN) ו־ENTER. כעת התכנית אמורה לפעול.

פוליהדרה

```
  5 BORDER 1: PAPER 6: CLS
 10 INPUT n
 20 FOR r=20 TO 80 STEP 2
 25 LET x=128: LET y=87
 30 LET h1=x-r: LET v1=y
 35 PLOT h1,v1
 40 FOR a=0 TO 361 STEP 360/n
 50 LET h2=x-r*COS (a*PI/180)
 60 LET v2=y+r*SIN (a*PI/180)
 70 DRAW h2-h1,v2-v1
 80 LET h1=h2: LET v1=v2
 85 BEEP 0.02,r-20
 90 NEXT a
100 NEXT r
```



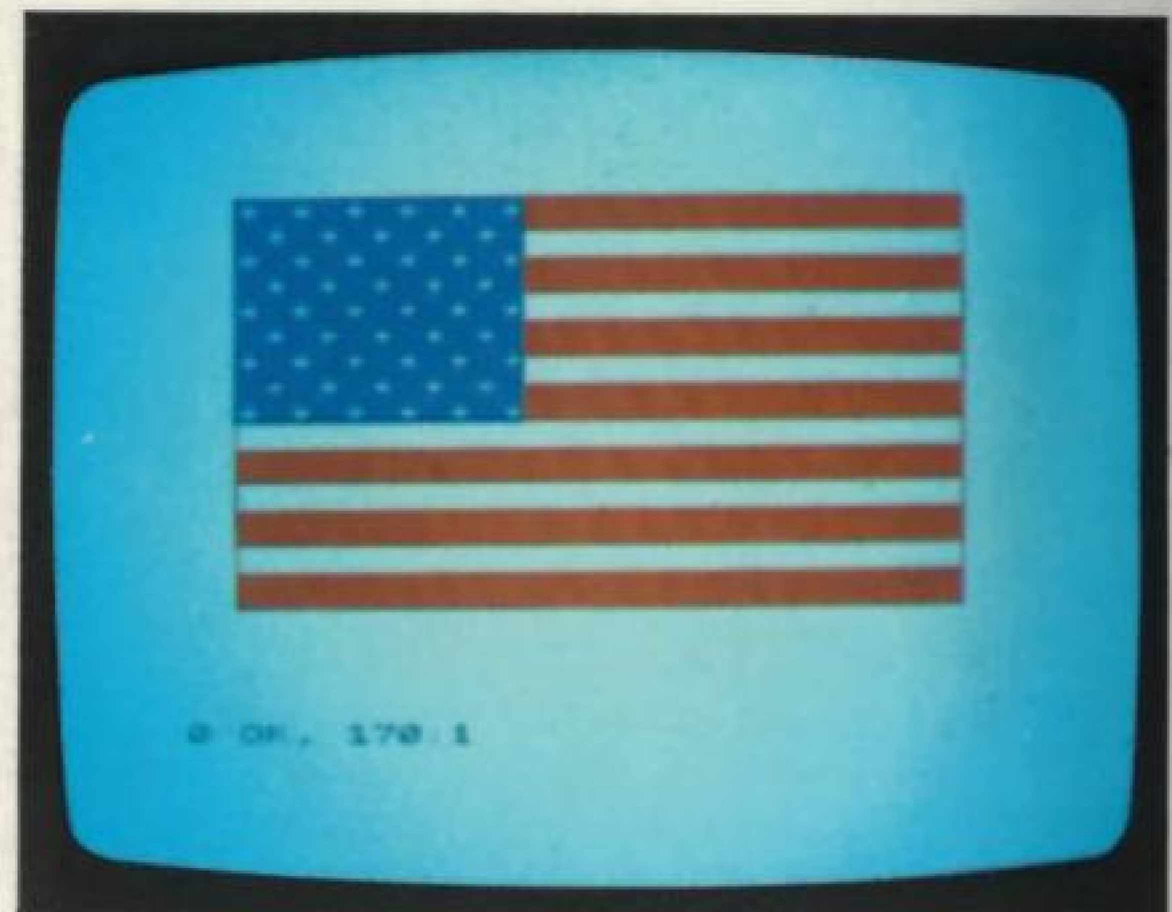
בתחילה, אתה רואה מסך ריק. הקש 6, ולחץ ENTER. צורה בת 6 צלעות תיבנה על המסך, התחל שנית את התכנית כאשר הסתיימה והקש מספר אחר לקבלת צורה עם מספר צלעות שונה.

נסה זאת

בשורה 20, שנה 2 למספר אחר. הצורה נבנית מהר יותר אם המספר גדול יותר, והפוליהדרה (צורות בעלות משטחים רבים) מרוחקות יותר זו מזו.

פסים וכוכבים

```
10 INK 2
20 PAPER 7
30 CLS
40 FOR x=28 TO 148 STEP 20
50 FOR z=0 TO 11
60 PLOT 16,z+x: DRAW 216,0
70 NEXT z
80 NEXT x
90 PLOT 16,28: DRAW 0,131
100 PLOT 232,28: DRAW 0,131
110 PAPER 1
120 INK 7
130 FOR x=2 TO 8 STEP 2
140 PRINT AT x,2;"* * * * * *"
150 PRINT AT x+1,2;" * * * * *"
160 NEXT x
170 PRINT AT x,2;"* * * * * *"
```

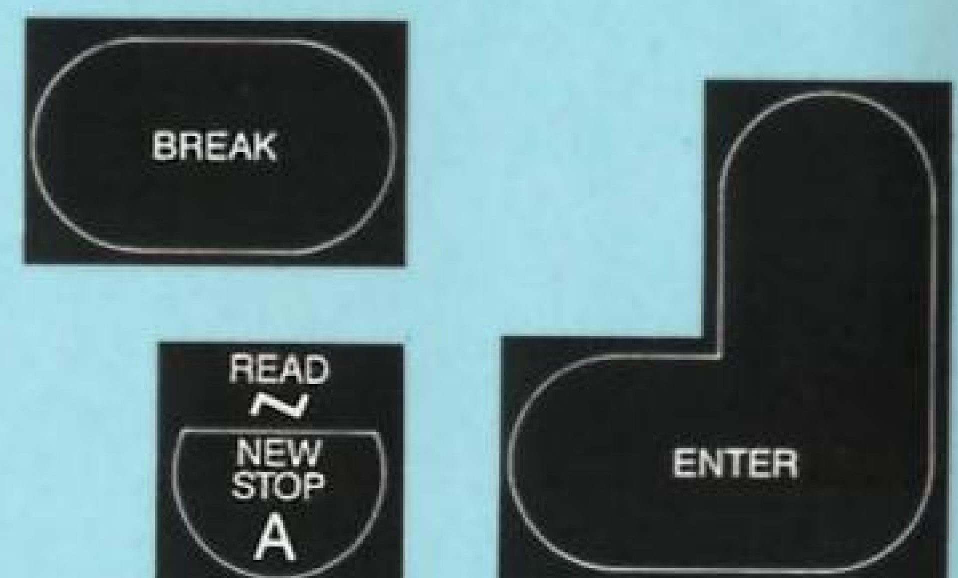


על המסך יופיע דגל ארה"ב.

נסה זאת
שנה את מספרי הצבע בדגל. צבע הפסים נמצא בשורה 10, הכוכבים בשורה 120 והרקע לכוכבים בשורה 110.

כיצד להתחיל תכנית חדשה

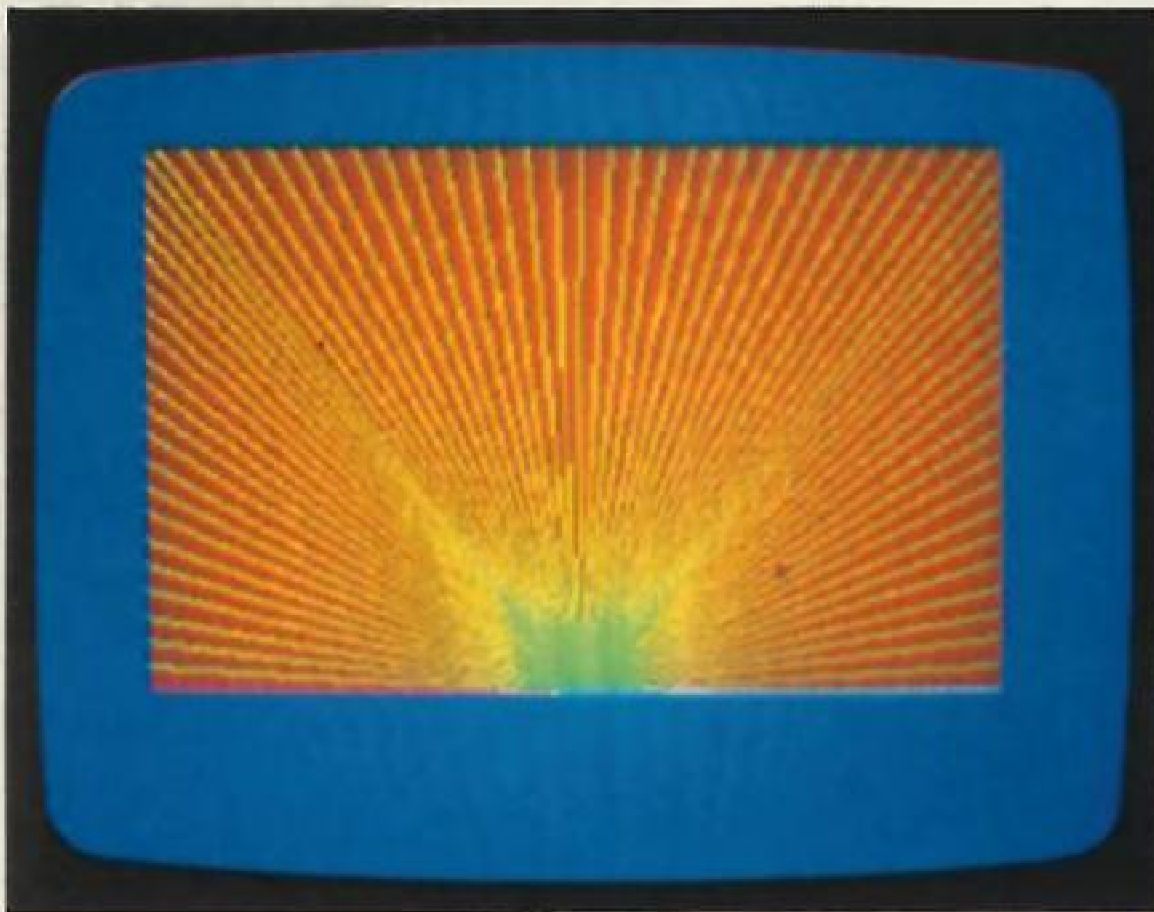
כאשר סיימת לעבוד עם תכנית מסויימת וברצונך להכנס לתכנית חדשה לגמרי, המתן עד סיום התכנית הנוכחית או עצור אותה בלחיצת BREAK. במצב זה תוכל לבחור באחת משתי דרכים למחיקת התכנית הישנה מזכרון המחשב. דרך אחת היא ללחוץ על שני מקשים, A(NEW) ואז ENTER. המסך ישחיר לרגע, ואז תופיע הודעת 'זכויות יוצרים'.



לחילופין, ובצורה קלה יותר, תוכל פשוט ללחוץ על כפתור RESET. לכפתור זה יש אותה השפעה כמו כיבוי הספקטרום ע"י ניתוקו מאספקת הזרם וחיבורו שנית.

זריחה מנצנצת

```
10 BORDER RND*6
20 INK RND*7
30 PAPER RND*6
40 CLS
50 LET z=RND*10+2
60 FOR x=0 TO 174 STEP z
70 PLOT 128,0
80 DRAW -128,x
90 BEEP .01,x/3
100 NEXT x
110 FOR x=-127 TO 127 STEP z
120 PLOT 128,0
130 DRAW x,175
140 BEEP .01,60
150 NEXT x
160 FOR x=174 TO 0 STEP -z
170 PLOT 128,0
180 DRAW 127,x
190 BEEP .01,x/3
200 NEXT x
210 PAUSE 200
220 GO TO 10
```



תמונה הדומה לזריחה מנצנצת בשלל צבעים נבנית מחדש בכל כמה שניות. אם המסך מתרוקן, פשוט המתן. בקרוב תראה שחר חדש מפציע.

נסה זאת
בשורה 210, שנה 200 למספר אחר במטרה לשנות את פרק הזמן בו נשארת כל זריחה על המסך. 200 שווה ל־4 שניות.

מה הלאה?

כעת יש לך ברירה. אם ברצונך לשמור חלק מתכניות אלה כדי להריצן שנית בעתיד, אתה יכול להקליטן על קלטות. עיין בפיסקה 'כיצד תשמור את תכניותיך' בעמ' 38 כדי ללמוד כיצד עושים זאת.

אם ברצונך להמשיך בניסויים עם הספקטרום, תוכל ללמוד על תכנות בפרק 2 — צעדים ראשונים בתכנות. עד עתה רק ניסית תכניות כתובות, בלי להבין בהכרח כיצד הן פועלות. פרק 2 יסביר כמה מתכונות התכנות של הספקטרום.

אם ברצונך לנסות כמה סרטי תכנה, כמו משחק מחשב שקנית, עיין בפיסקה הבאה 'כיצד להשתמש בתכנה מוכנה לריצה'.

כיצד להשתמש בתכנה מוכנה לריצה

כאשר הכנסת תכנית לספקטרום, יצרת רצף של אותות מקודדים בצורה אלקטרונית כשלחצת על המקשים. הקודים מופנים לזכרון הספקטרום, אשר אוגר אותם כדי שהמחשב יוכל להשתמש בהם כאשר התכנית רצה. הקודים נשארים בזכרון עד שאתה מסיר אותם (ע"י הקשת NEW או לחיצת כפתור RESET, לדוגמה) או מכבה את הספקטרום.

ברם, אין זה תמיד הכרחי להקיש פנימה תכנית כאשר ברצונך להשתמש בספקטרום. במקום זאת, אתה יכול לרכוש תכנה מוכנה לריצה, המכילה תכניות שניתן להזין ישירות למחשב באופן אוטומטי. השימוש בתכנה מוכנה לא רק חוסך לך את זמן ההקשה בכל פעם שתפעיל את הספקטרום, אלא מאפשר גם בניית ספריית תכניות מוכנות לשימוש, אשר היו דורשות ימים ואף שבועות של עבודה אילו היית כותב אותן בעצמך. יצרני תכנה מייצרים תכניות מכל הסוגים, אשר נכתבות בידי מתכנתים מעולים, ועבור הספקטרום מוצע מגוון עצום של תכניות. עיין בקטלוג התכנה של ספקטרום כדי לקבל מושג באילו תכניות תוכל להשתמש. ואז ברגע שתרצה, תוכל להריץ תכנית המתאימה לצרכיך.

כיצד לטעון תכניות

אותות הקוד על סרט תכנה מכילים סדרת צפצופים גבוהים ונמוכים המוקלטים בקצב של כ־1500 בשנייה. כאשר אתה משמיע סרט תכנה ברשמקול, המכשיר מייצר את רצף הצפצופים המהווים את התכנית. כל שעליך לעשות הוא לחבר את הרשמקול לספקטרום, והקודים יעברו ישירות לזכרון המחשב. תהליך זה נקרא טעינת תכנית.

בפרק זה תלמד כיצד לחבר את רשמקול הקלטות. עמ' 14-15 מתארים כיצד להשתמש בו.

שאלות ותשובות בנושא תכנה

מה פירוש המונח 'תכנה'?
תכנה היא השם הכללי הניתן לתכניות המוזנות למחשבים כדי שיפעלו. חמרה היא המונח למכונות הממשיות – המחשבים עצמם וכל התקן אחר המעורב במיחשוב.

מדוע תכנה מיוצרת על סרטי קלטות?
סרטי קלטות נוחים לשימוש ואינם דורשים ציוד מיוחד. כל מה שאתה זקוק לו לצורך טעינת סוג תכנה זה הוא רשמקול קלטות רגיל.

כיצד נשמעות תכניות מוקלטות על סרט?
השמע תכנית מוקלטת ברשמקול ללא חיבורו לספקטרום. אתה תשמע צפצוף צפצוף חד. הצפצוף נגרם עקב אותות הקוד אשר מופנים לרמקול הנמצא ברשמקול במקום למחשב. האותות נשלחים מהקלטת למחשב במהירות כה גבוהה עד כי לא ניתן להבחין בין הקולות הבודדים.

האם קיימים סוגי תכנה אחרים?
כן. ניתן להשיג תכניות מוקלטות על מחסניות רום במקום קלטות. המחסנית מתחברת לממשק אשר מתאים למחבר בחלק האחורי של הספקטרום. תכנית במחסנית רום נטענת מיידית ללא כל המתנה.

כמו כן ניתן להשיג תכנה על תקליטון כונן זעיר. התקליטון מכיל תכניות שהוקלטו בצורה מגנטית כמו על סרט קלטת. כמה תכניות יכולות להמצא יחד על התקליטון, ולא כמו בסרט קלטת, כל תכנית ניתנת לטעינה תוך שניות במקום דקות. תקליטונים מופעלים בעזרת יחידת כונן זעיר (ראה עמ' 46).

מהו הרשמקול הטוב ביותר?
הספקטרום יפעל היטב עם כל רשמקול נייד זול, רצוי כזה שמחובר לרשת החשמל ולא לסוללות. למכשיר צריכה להיות בקרת עצמת קול עצמאית אך אין הכרח בבקרת טון. קיימים גם רשמקולים מיוחדים למחשבים. אלה מתוכננים לאגירה וטעינת תכניות בצורה אמינה יותר ממכשירים רגילים. נגן קלטות (Cassette deck) המהווה חלק ממערכת שמע גדולה מסובך יותר לחיבור למחשב. בנוסף לכך, שקעי תפוקת הקול בנגן קלטות אינם מפיקים אותות חזקים מספיק לספקטרום.

האם תכניות מוקלטות דורשות טיפול מיוחד?
כמו כל צורה אחרת של איחסון מגנטי, תכניות על קלטות עלולות להנזק משדות מגנטיים חזקים, לכן אל תאחסן את הקלטות ליד מכשיר הצורך זרם חשמלי חזק. קלטות תכנה חייבות גם להשמר במקום נקי מאבק.

האם כל סוג של תכנה יפעל?
לא. רק תכנה המיוצרת עבור הספקטרום ZX או ספקטרום ZX + ניתנת לטעינה.

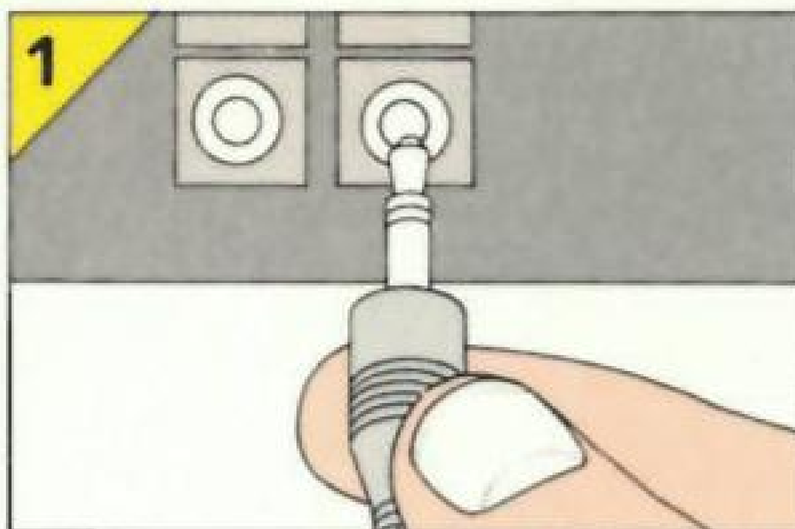
כיצד לחבר רשמקול קלטות

כבל הרשמקול המסופק עם הספקטרום מיועד לחיבור רשמקול קלטות. זהו כבל עם זוג תקעים קטנים בכל קצה. הצב את הרמקול ליד הספקטרום וחבר את הכבל כמתואר בציורים. הרשמקול והספקטרום יכולים להיות מופעלים או מכובים בזמן החיבור, למרות שכדאי להוציא את הקלטת מהרשמקול לפני הפעלתו או כיבויו. כך תשמור על התכניות המאוחסנות בקלטת.

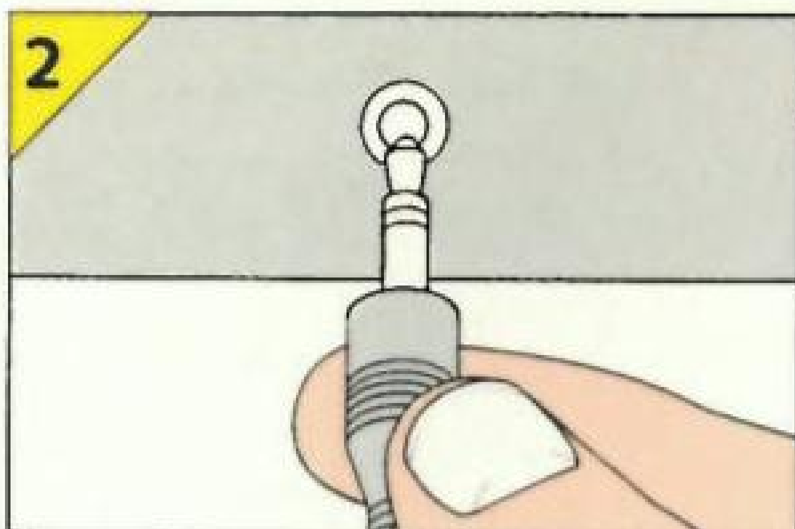
כיצד לחבר נכון

עצות תכנה

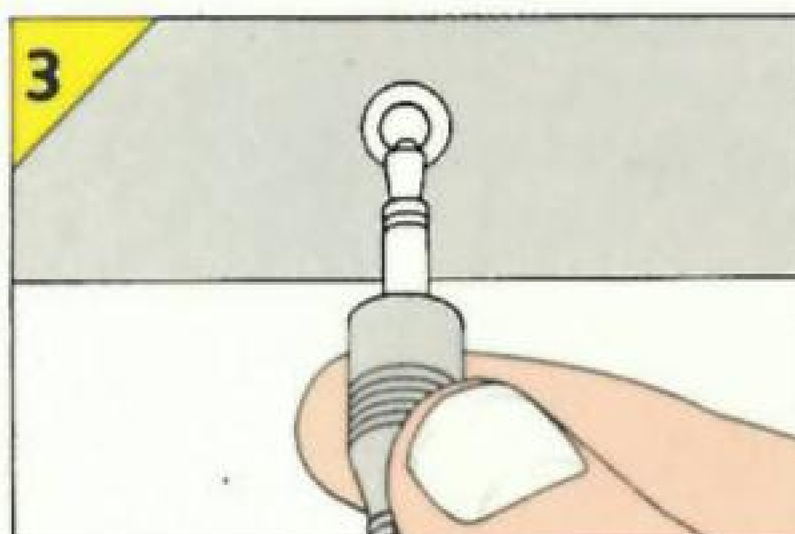
- כבל הקלטת בספקטרום כולל תקעים צבעוניים כדי למנוע חיבורים מוצלבים מוטעים בין שקעי המחשב והרשמקול. כאשר אתה משתמש ברשמקול עם הספקטרום, השתדל תמיד להצמד לאותה מערכת, עם צבע אחד לשקעי EAR והאחר לשקעי MIC.
- סוגי רשמקולים מסויימים עלולים להיות מושפעים מציוד חשמלי סמוך. לפעמים הדבר עלול לגרום לעיוות האותות הנשלחים בין המחשב לרשמקול וכתוצאה מכך התכניות לא ייטענו היטב. אם הרשמקול לא פועל היטב לעיתים קרובות למדי, נסה להעבירו כך שלא יהיה ליד מקלט הטלביזיה או המחשב.



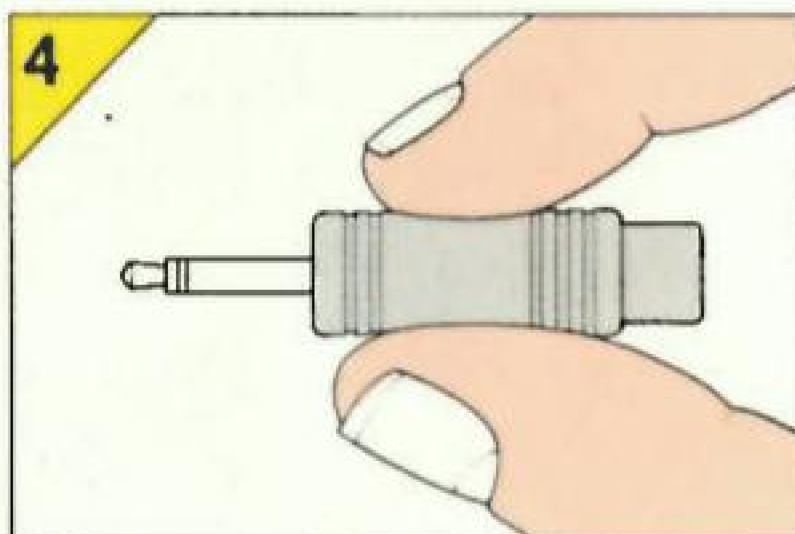
הכנס אחד מארבעת התקנים לשקע EAR בחלק האחורי של הספקטרום.



הכנס את התקע השני באותו צבע לשקע EAR ברשמקול, אם הוא קיים.



אם אין שקע EAR ברשמקול, חבר את התקע לשקע אזניות אם הוא קיים. במקרה אחר, נסה לחברו לשקע רמקול חיצוני.



אם התקע בכבל הקלטת אינו מתאים לשקע ברשמקול, תזדקק למתאם או כבל מיוחד עם התקעים המתאימים שתוכל להשיג בחנות לדברי אלקטרוניקה. שקע EAR בספקטרום דורש תקע פין 3.5 מ"מ ואות קלט של כ־1 וולט.

שקעי EAR ו־MIC

כאשר טוענים תכניות אתה יכול לחבר את שקעי EAR ו־MIC, כמתואר כאן. אך כאשר אתה מקליט תכניות (ראה עמ' 38) **חובה** לנתק כבל EAR.



כיצד לטעון תכנית

כעת, לאחר שחיברת את הרשמקול לספקטרום, אתה מוכן לטעון ולהריץ תכנית. אתה יכול להשתמש בסרט תכנה מוכנה לריצה או בסרט המכיל את התכניות שכתבת. הנוהל זהה בשני המקרים.

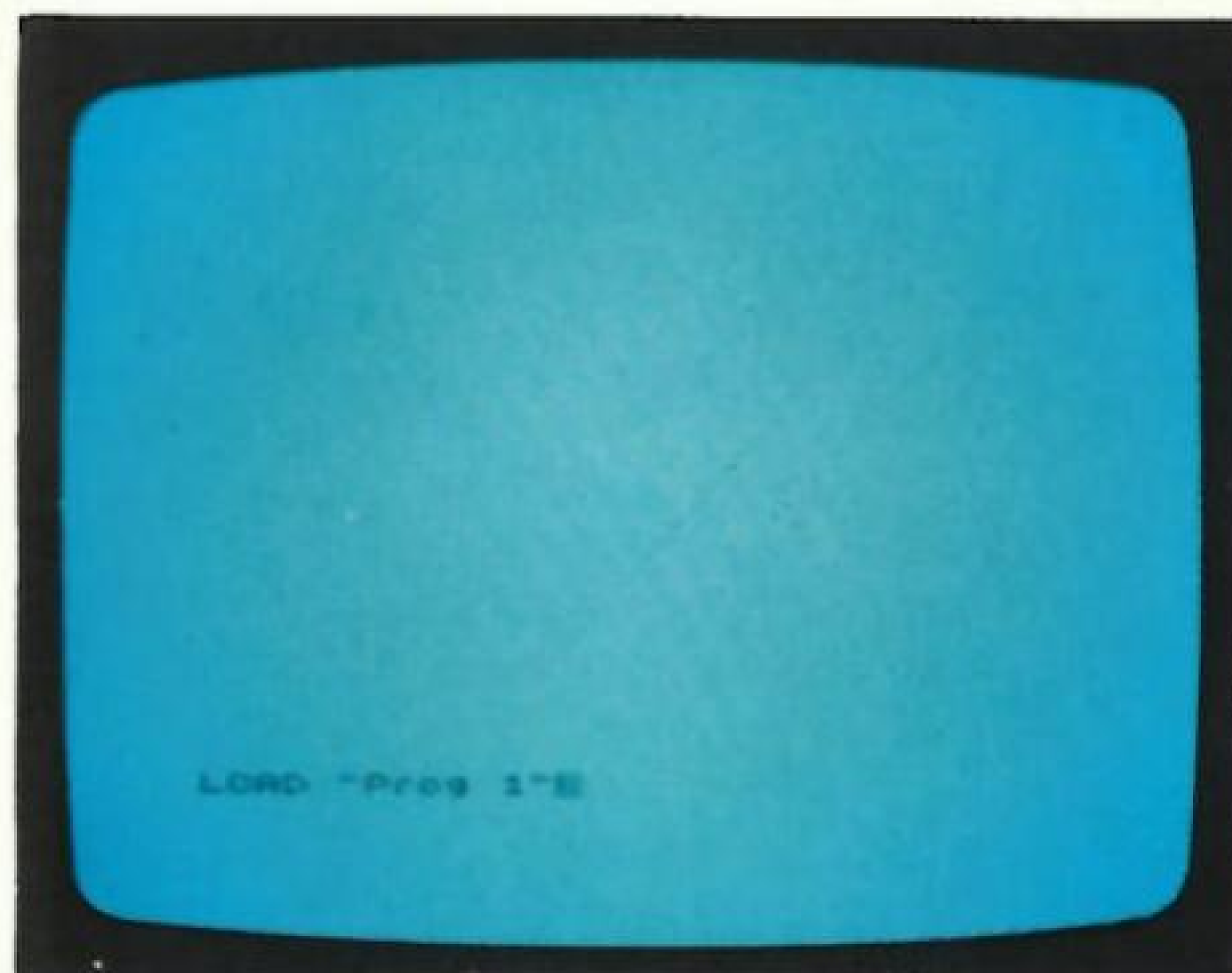
הפעל את רשמקול הקלטות. וודא שהספקטרום מופעל, והכנס את הקלטת לרשמקול. אם קיימת כבר תכנית במחשב, המתן עד שתסתיים או שתוכל לעצרה בלחיצת מקש BREAK. כעת תוכל להקיש NEW או ללחוץ על כפתור RESET כדי למחוק את התכנית מזכרון הספקטרום, אך אין זה הכרחי מכיוון שטעינת תכנית חדשה מנקה תחילה את הזכרון. חשוב לזכור שאם אתה טוען תכנית, התכנית הקודמת תמחק מהזכרון.

כעת עקוב אחר ההוראות הממוספרות. אם משהו אינו כשורה, עיין במאתר תקלות טעינת תכנה, בעמ' 16.

1 הכנס את הקלטת וגלגל אותה לנקודת ההתחלה

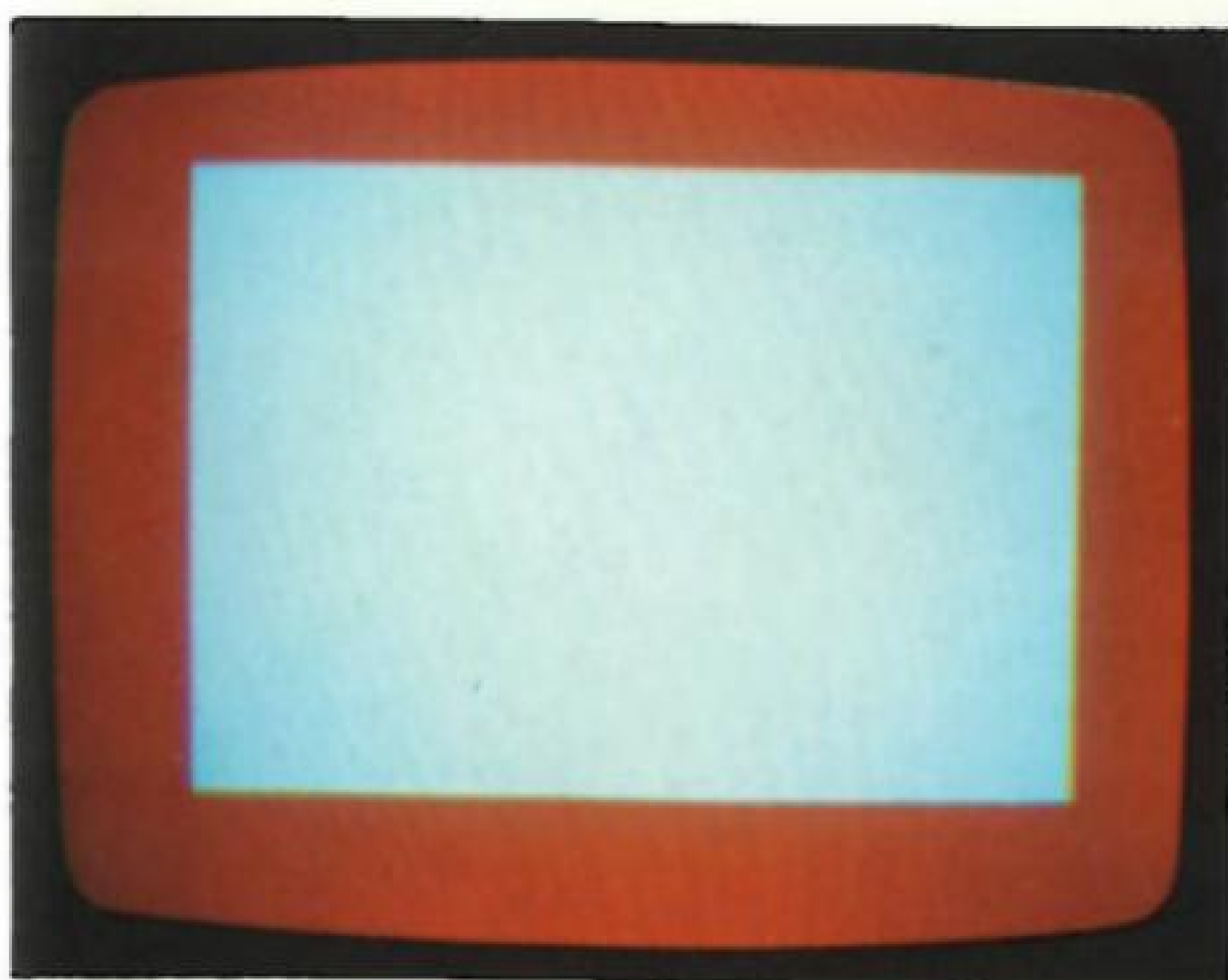
2 הצב את פקדי עצמת קול ובקרת טון של הרשמקול ברמות הדרושות. נסה להציב עצמת קול בערך של 2/3 מעוצמה מירבית ואם יש בקרת טון, הצב אותה ב־treble מירבי.

3 לחץ על J ועל המסך יופיע LOAD. כעת הקש את שם התכנית בין מרכאות כפולות. לדוגמא, LOAD "PROG 1"



4 לחץ ENTER, המסך יתרוקן.

5 הפעל את הסרט. גבול המסך יקבל צבע אדום או כחול או הבזקי אדום וכחול. תופעה זו מציינת שהספקטרום מחפש תכנית.



6 לאחר מספר שניות, יתחילו לנוע מעלה ומטה באיזור גבול המסך פסים אדומים וכחולים. זה מציין שהספקטרום התחיל לקלוט אותות.

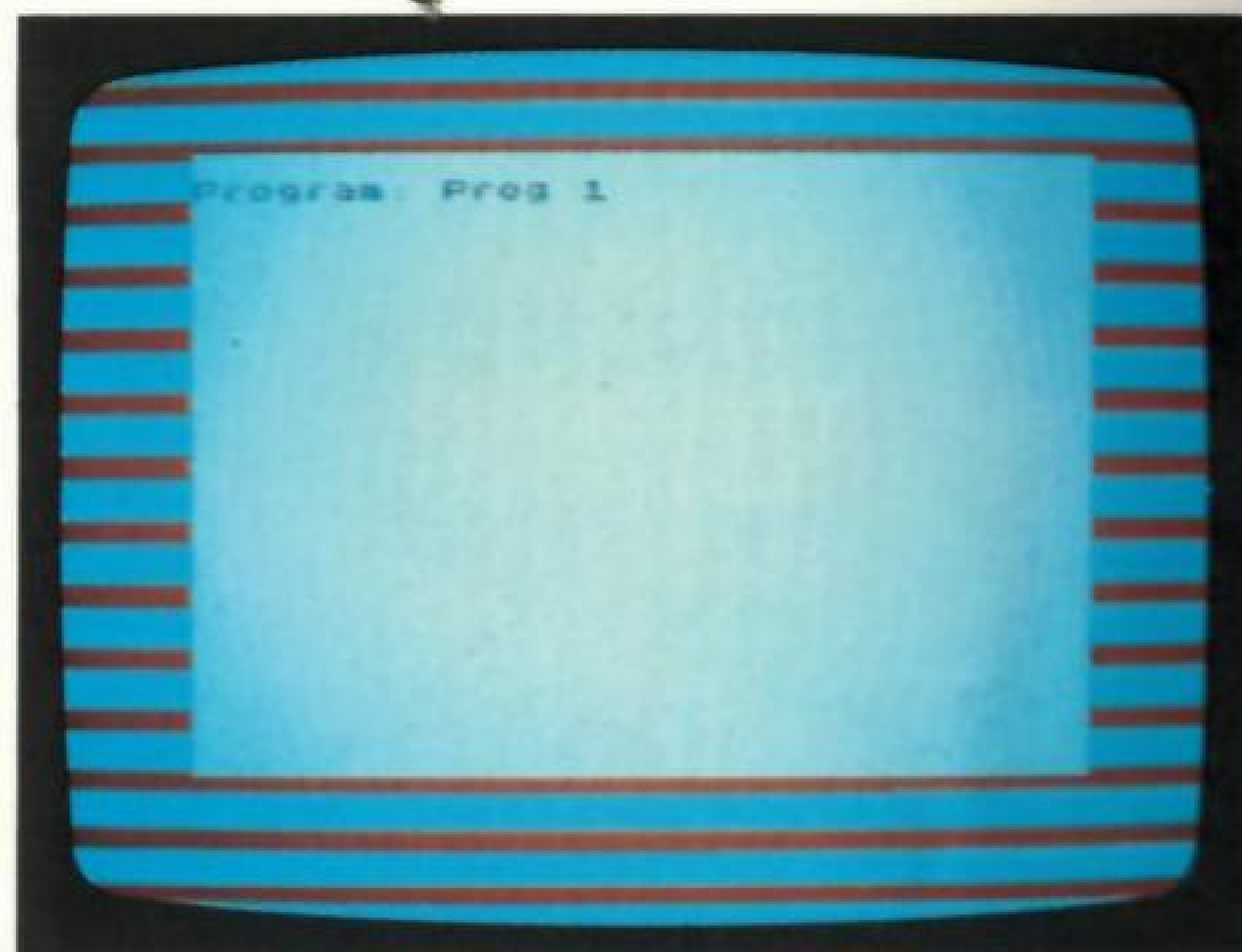
עצות לטעינת תכנה

להלן מספר עצות שיעזרו לך לחסוך זמן בעת הטעינה.

1 כתוב תוויות זיהוי לכל הקלטות כך שתוכל למצוא בקלות את התכניות. אם סרט מכיל יותר מתכנית אחת, כתוב את שמות התכניות לפי הסדר על התוית. זכור לאיית את שם התכנית בדיוק כפי שהמחשב ידרוש.

2 אם לרשמקול יש מונה, העזר בו למציאת תכנית במהירות על סרט המכיל יותר מתכנית אחת בכל צד. אפס את המונה בתחילת הסרט, ואז הקש LOAD בלווי שם תכנית כלשהי (במרכאות) שאיננה על הסרט. השמע את הסרט והספקטרום יתן שם לכל תכנית שהוא מוצא, בלי לטעון אותה. כתוב את מספרי המונה על התוית בצד שם התכנית. מאוחר יותר, תוכל להגיע במהירות ובקלות לתכנית הדרושה.

7 המילה **Program** בלווי שם התכנית, או: **Bytes** בלווי שם או אות, יופיעו על המסך. זה מציין שהמחשב איתר בהצלחה את התכנית.



8 הפסים האדומים וכחולים מופיעים שנית כאשר המחשב ממתין לטעינר התכנית.

9 בגבולות המסך מופיע רצף של פסים צהובים וכחולים. זה מציין שהספקטרום טוען את התכנית. טעינה יכולה לארוך מספר דקות אם התכנית ארוכה מאד.



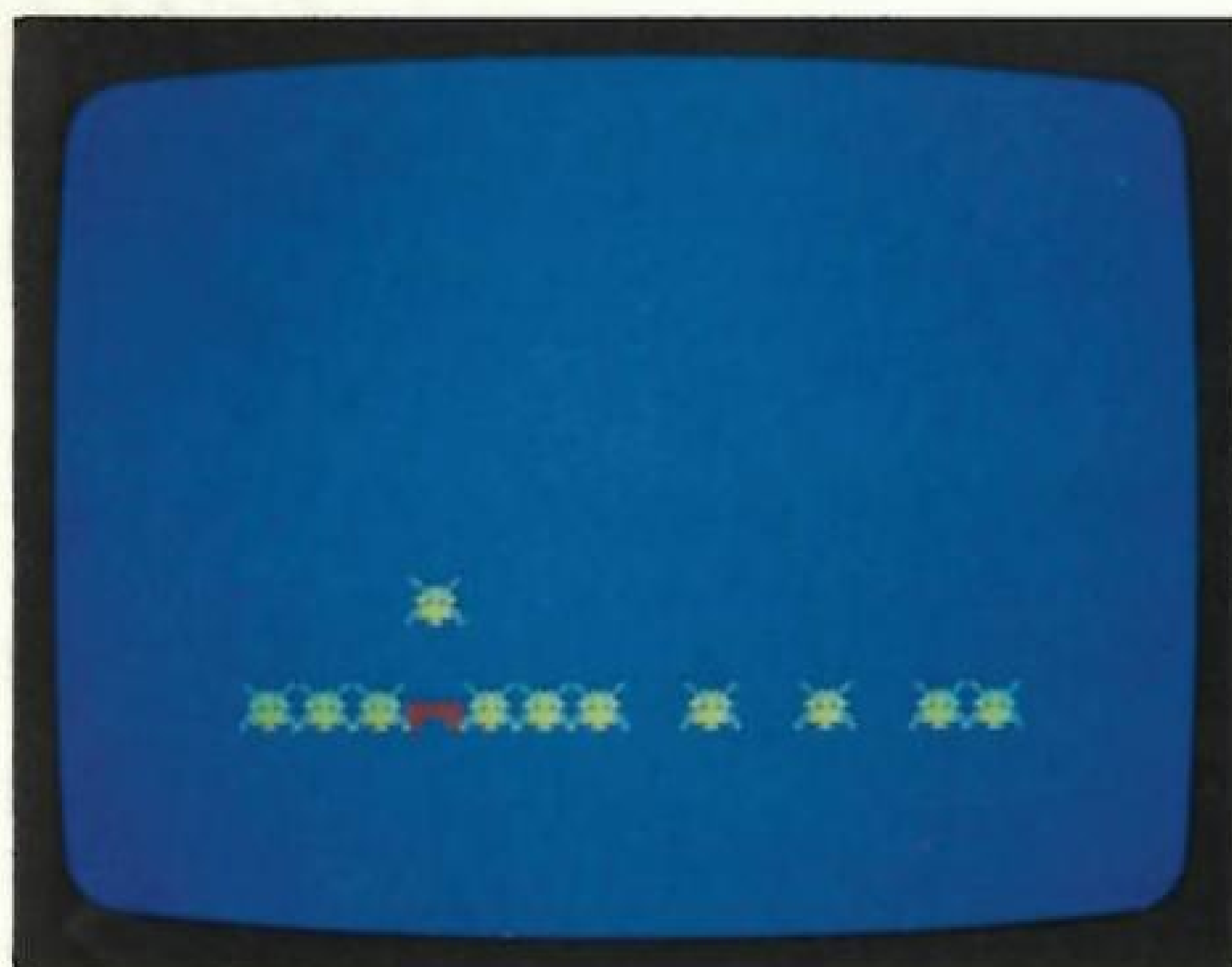
10 פעולות 7, 8 ו־9 עשויות לחזור מספר פעמים אם התכנית מפוצלת לכמה חלקים.

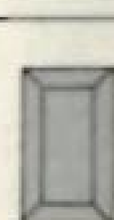
11 התכנית עשויה להתחיל באופן אוטומטי עם גמר הטעינה. זכור לעצור את הסרט.

12 אם התכנית לא מתחילה לרוץ באופן אוטומטי כאשר נטענה, המסך יתרוקן ויופיע הדווח 0:1 ,0 OK. עצור את הסרט.

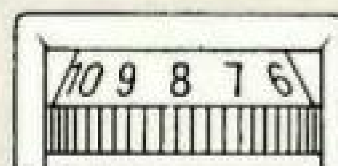


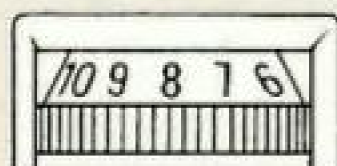
13 לחץ R(RUN) ו־ENTER, והתכנית תתחיל כעת לרוץ.





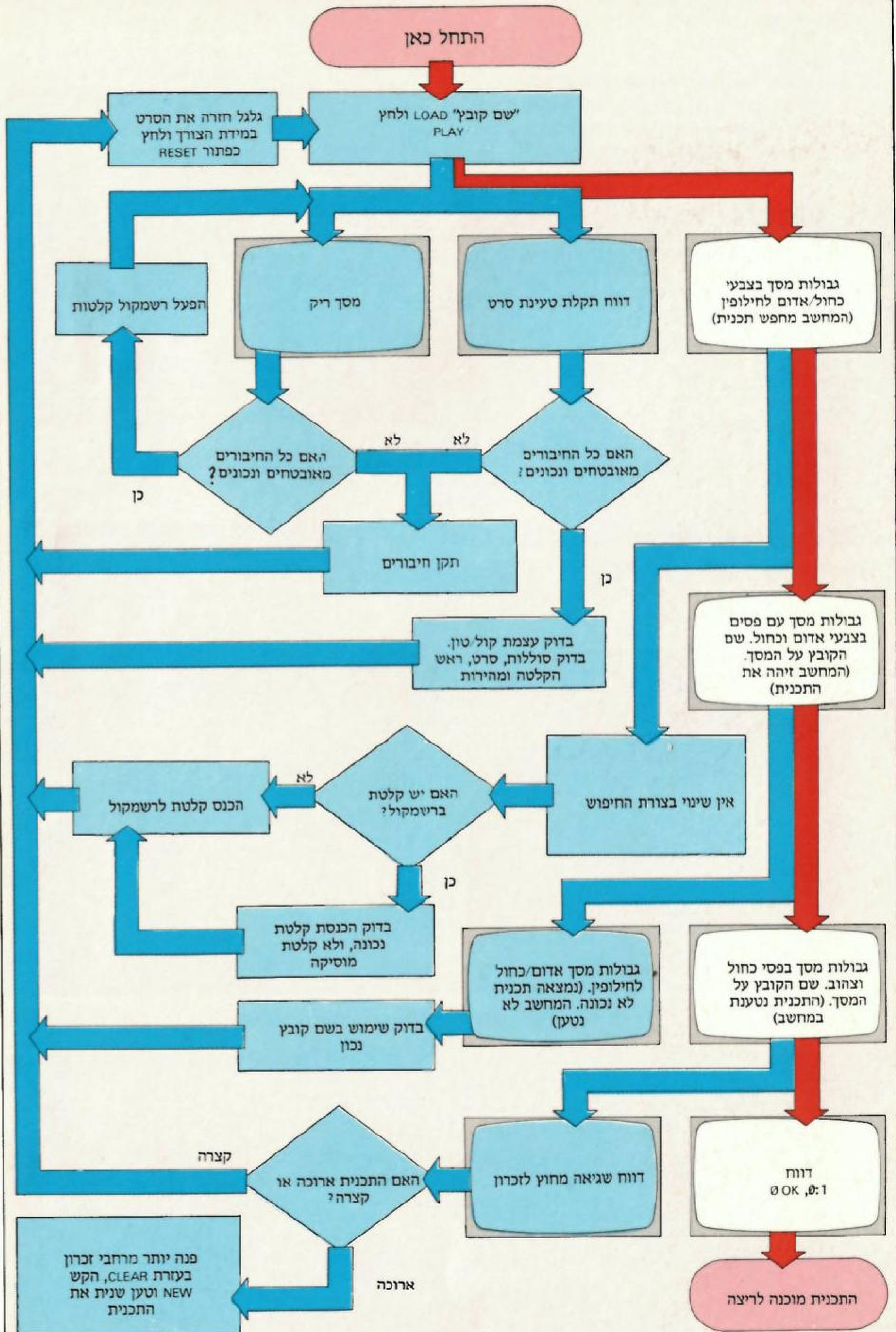
3 אם הסרט נמצא בתכנית הנכונה או אם אינך יודע את שם התכנית, הקש "" LOAD במקום LOAD המלווה בשם התכנית במרכאות. אסור שיהיה רווח בין שני סימני המרכאות. הספקטרום יטען את התכנית הראשונה שימצא בסרט. אם שם התכנית המופיע אינו זה שרצית, לחץ BREAK, גלגל הלאה ונסה שנית.





4 שים לב לרמות עצמת הקול ובקרת טון המאפשרות לספקטרום טעינת תכניות. הצב רמות אלה ברשמקול לפני הטעינה.

איתור תקלות טעינת תכנה



צעדים ראשונים בתכנות

פרק זה מהווה הקדמה לכתיבת תכניות עבור הספקטרום +ZX. הפרק מסביר כיצד תוכל להכיר את הספקטרום דרך המקלדת. אח"כ תראה כיצד תתחיל להפעיל את הספקטרום לצרכיך. התכניות הקצרות המובאות כאן עבורך מתרכזות בתכונות המיוחדות של הספקטרום, כך שכאשר תתחיל בכתיבת תכניות משלך תוכל לנצל את מלוא יכולתו של המחשב.



המקלדת — לוח הבקרה של המחשב

לספקטרום +ZX יש שפה משלו, שפת המחשב הידועה בשם בייסיק. כדי שהמחשב יציית להוראותיך, עליך לתכנת את הספקטרום תוך דיבור אליו בשפת בייסיק. אתה עושה זאת באמצעות המקלדת. יתרה מזאת, המקלדת מאפשרת לך לבקר את המחשב בזמן שהוא מריץ את תכניותיך.

הניב (הגירסה) של בייסיק אותו מבין הספקטרום הוא צורה פשוטה של שפה זו אך רבת עצמה. היא מתוכננת להיות דומה עד כמה שיותר לשפה האנגלית כדי להקל על העבודה. בנוסף לכך, לספקטרום יש תכונה אחת גדולה המקלה מאד על התכנות. זוהי שיטת הכנסת מילות מפתח בהקשת מקש אחת.

מקשים ומילות מפתח

מילות מפתח הן מילים מיוחדות בשפת בייסיק המורות למחשב לבצע משהו — מילים כמו PRINT ו־INPUT. במרבית המחשבים עליך להקיש כל אות של מילת מפתח כפי שנעשה במכונת כתיבה, ועליך לאיית במדוייק כל מילה. לעומת זאת בספקטרום אתה לוחץ על מקש אחד בלבד לקבלת מילת מפתח מלאה על המסך.

מקשי מספרים — בנוסף ליצירת מספרים, מקשים אלה יכולים לייצר קודי בקרה בתכניות עבור הצבעים המתוארים — ראה עמ' 33. מילות המפתח הנמצאות מייד מעל ראשי המקשים מ־4 עד 0, פרט למקש 8, משמשים לבקרת הכונן הזעיר ZX בלבד.

מקשי תצוגת צבע — ששה מקשים אלה מייצרים מילות מפתח המבקרות את דרך הצגת הצבע על המסך.



BREAK — מקש זה עוצר ריצת תכנית. הוא לא מוחק את התכנית מהזכרון.

ENTER — לחץ על מקש זה כדי להכניס שורת תכנית לזכרון הספקטרום. משמש גם להזנת מידע למחשב בזמן תכנית.

SYMBOL SHIFT — החזק מקש זה לחוץ והקש אות או מספר לבחירת מילת המפתח או הסימן התחתונים בחלק המורם של כל מקש. כאשר הוא נלחץ אחרי EXTEND MODE הוא בוחר את הסימן או מילת מפתח הנמצאים מייד מעל החלק המורם של המקש — ראה עמ' 20-21.

מקש רווחים — מקש זה מייצר רווח בדומה למקש הרווחים במכונת כתיבה.

בקרי סמן — לחיצה על מקשים אלה גורמת לסמן לנוע בכוון החיצים. מקשים אלה נמצאים לרוב בשימוש תכניות לבקרת תנועת צורות על המסך. כמו כן משתמשים בהם לצורך עריכת תכניות.

בבייסיק של סינקלייר יש למעלה מ־80 מילות מפתח, הנכתבות בעזרת 36 מקשים בסך הכל (26 מקשי אותיות ו־10 מקשי מספרים). מכיוון שהספקטרום משתמש במבחר גדול כל כך של פקודות בייסיק, מקשים רבים מייצרים לא אחת כי אם כמה מילות מפתח שהמחשב מכיר. מרבית המקשים נותנים למעשה מילות מפתח יחד עם אות, מספר, סימן או אפילו צורה (תו גרפי), אשר ניתן להשתמש בהם בתכניות.

בחירת מילות מפתח וסימנים

במקלדת הבייסיק יש שני מקשים בעלי שימוש רב ביותר. אלה הם מקשי EXTEND MODE ו־SYMBOL SHIFT, אשר מאפשרים לך לבחור מי מתוך כל מילות המפתח והסימנים שעל מקשים אחרים יופיעו על המסך. היתה לך כבר התנסות קצרה בהקשה על מקשים אלה בעמ' 8. כעת, לאחר שתרגלת את עצמך בהכרת המקלדת, שני העמודים הבאים יראו לך כיצד לבחור כל מה שמופיע על המקלדת. לאחר שתכיר את כל המקשים, תוכל להתחיל לכתוב תכניות בעצמך.

GRAPH – מקש זה משמש לבחירת הצורות או תווים גרפיים על מקשים 1 עד 8. אם אתה לוחץ על מקש זה ואח"כ לוחץ על מקש מספר עם או בלי מקש CAPS SHIFT, יופיע תו גרפי על המסך.

TRUE VIDEO ו־**INV VIDEO** רק מקשים אלה מכניסים קודי בקרה לשורות תכנית כדי ליצור צבעים רגילים או הפוכים.

NEW – מקש זה מנקה את מרחב הזכרון של בייסיק במחשב, כשהוא מוחק כל תכנית הנמצאת שם.

DELETE – במקש זה תשתמש אם לחצת על מקש לא נכון וברצונך למחוק מילת מפתח, אות, מספר או סימן – ראה עמ' 10.

EDIT – מקש זה משמש לשינוי שורה בתכנית ללא צורך לכתבה שנית בשלמותה – ראה עמ' 21.



EXTEND MODE – מקש זה בוחר את מילת המפתח העליונה מעל החלק המורם בכל מקש. כאשר הוא מלווה ב־SYMBOL SHIFT ומקש, הוא בוחר את הסימן או מילת המפתח הנמצאים מייד מעל החלק המורם של המקש – ראה עמ' 20-21.

CAPS SHIFT – לחץ על מקש זה עם מקש אות ליצירת אות גדולה. אם אתה רוצה מספר אותיות גדולות, השתמש ב־CAPS LOCK.

CAPS LOCK – השתמש במקש זה אם תרצה שמקשי האותיות יצרו אותיות גדולות כל הזמן. לחץ שנית לקבלת אותיות קטנות.

BEEP מקש זה מייצר את מילת המפתח המבקרת את הסינתסייזר הקולי של הספקטרום.

כיצד להפעיל את המקשים

אתה יכול לקבל עד 6 מילות מפתח, אותיות, מספרים או סימנים שונים ממרבית המקשים בספקטרום +ZX. ברם, בחירת אות או מילת מפתח על המקלדת אינה כה מסובכת לאחר שהכרת את אחת מתכונותיו המיוחדות של הספקטרום. אם אתה לוחץ על מקש, התוצאה שתופיע על המסך תלוייה באופן הפעולה בו נמצא המחשב באותו זמן. כל אופן פעולה מאפשר הקשת סוגי מידע אחרים, כמו מילות מפתח, אותיות או תווים גרפיים. היתרון בכך הוא, שבזמן הפעלת המקלדת, הספקטרום בעצם עוזר לך בבחירת אופני פעולת המקלדת כך שתקיש פקודות ונתונים בסדר הנכון. בשני עמודים אלה תמצא מה בדיוק עושים כל אופני הפעולה.

אופן פעולה – מילת מפתח

הפעל או אתחל את הספקטרום כך שתופיע הודעת זכויות יוצרים. כעת לחץ ENTER. בפינה השמאלית תחתונה יופיע K מהבהב. הריבוע המהבהב נקרא סמן. הוא מראה לך היכן עתיד להופיע משהו על המסך והאות K מציינת שהמחשב נמצא באופן פעולה של מילת מיתח (keyword). לחץ על מקש אות כלשהו ומילת המפתח העליונה בחלק המורם של המקש תופיע על המסך. נסה Q, למשל, ומילת המפתח **PLOT** תופיע. לחץ על מקש DELETE למחיקת מילת המפתח ונסה מקשים אחרים. מקשי מספרים יתנו מספרים, אך אם תלחץ על מקש אות, תופיע מילת המפתח העליונה בחלק המורם של המקש.

השתמש שנית ב־DELETE כך שהסמן K יופיע שוב. כעת לחץ על אחד משני מקשי SYMBOL SHIFT, החזק אותו לחוץ ולחץ על מקש אות כלשהו. הפעם, יופיעו על המסך מילת המפתח או הסימן הנמצאים מעל האות בחלק המורם של המקש. במקש מספר, יופיע הסימן בצד ימין בחלקו המורם של המקש. כלומר, אופן הפעולה – מילת מפתח מתייחס לחלק המורם של המקש.

כיצד לבחור מילת מפתח, סימן או תו

כאן תוכל לראות כיצד לבחור כל מילח מפתח, סימן או תו על מקש או מקש מספר. בזמן בחירת פונקציית מקש, שים לב למקומה על המקש, ובעזרת שתי הדוגמאות שלהלן קבע אילו מקשים – אם קיימים – תצטרך לשנות לאופן הפעולה הנכון. התבונן תחילה בסמן על המסך כדי לזהות את אופן פעולת המחשב.

מקש מספר



אופן פעולה (K) מילת מפתח		
מקש בלבד		3
SYMBOL SHIFT	ומקש	#
אופן פעולה (E) מורחב		
EXTEND MODE	ומקש בלבד	magenta
EXTEND MODE אח"כ	SYMBOL SHIFT ומקש	LINE
אופן פעולה (L) אותיות		
מקש בלבד		3
SYMBOL SHIFT	ומקש	#
אופן פעולה (C) אותיות גדולות		
CAPS LOCK	ומקש בלבד	3
CAPS LOCK אח"כ	SYMBOL SHIFT ומקש	#
אופן פעולה (G) גרפיקה		
GRAPH	ומקש בלבד	▀
GRAPH אח"כ	CAPS SHIFT ומקש	▄

מקש אות



אופן פעולה (K) מילת מפתח		
מקש בלבד		BORDER
SYMBOL SHIFT	ומקש	*
אופן פעולה (E) מורחב		
EXTEND MODE	ומקש בלבד	BIN
EXTEND MODE אח"כ	SYMBOL SHIFT ומקש	BRIGHT
אופן פעולה (L) אותיות		
מקש בלבד		b
CAPS SHIFT	ומקש	B
SYMBOL SHIFT	ומקש	*
אופן פעולה (C) אותיות גדולות		
CAPS LOCK	ומקש בלבד	B
CAPS LOCK אח"כ	SYMBOL SHIFT ומקש	*
אופן פעולה (G) גרפיקה		
GRAPH	עם מקשים A עד U בלבד עם גרפיקת משתמש.	

אופני הפעולה – אותיות ואותיות גדולות

לאחר יצירת מילת מפתח או סימן באופן הפעולה – מילת מפתח, המחשב משנה אוטומטית את הסמן ל־L. הסמן נמצא כעת באופן פעולה – אותיות. לחץ על מקש אות כלשהו והאות הקטנה תופיע. לחץ על מקש מספר והמספר יופיע. אם ברצונך להקיש אות גדולה, החזק מקש CAPS SHIFT לחוץ ולחץ על מקש האות.

אם ברצונך להקיש רק אותיות גדולות, לחץ CAPS LOCK תחילה. הסמן עובר למצב C. הספקטרום נמצא עתה באופן פעולה – אות גדולה ואתה תקבל רק אותיות גדולות על המסך. במצב אותיות גדולות ניתן עדיין לקבל מספרים. כדי לחזור לאופן פעולה – אותיות (L), לחץ שנית על CAPS LOCK.

אופן פעולה – מורחב (E)

אופן הפעולה – מורחב נוצר ע״י לחיצת מקש EXTEND MODE. הסמן משתנה עתה ל־E. לחץ על מקש אות כלשהו, ועל המסך תופיע מילת המפתח העליונה בזוג מילות המפתח שמעל החלק המורם במקש. לדוגמה, לחץ B ותקבל BIN. לקבלת מילת המפתח או הסימן התחתונים מעל חלק המקש המורם, לחץ על אחד ממקשי SYMBOL SHIFT תחילה, והחזק אותו לחוץ, יחד עם לחיצת מקש האות. על מקש B, לדוגמה, תקבל עתה BRIGHT כלומר, במצב פעולה – מורחב תקבל את זוג מילות המפתח מעל החלק המורם של המקש. לאחר לחיצת מקש (או EXTEND MODE) באופן פעולה – מורחב, המחשב חוזר אוטומטית לאופן פעולה של אותיות או אותיות גדולות.

אופן פעולה – גרפיקה

אופן הפעולה החמישי נקרא גרפיקה, ונוצר בלחיצת מקש GRAPH. הסמן משתנה ל־G. לחץ על מקשים 1 עד 8 וראה כיצד התווים הגרפיים המסומנים על מקשים אלה יופיעו על המסך. כעת לחץ CAPS SHIFT ומספר כלשהו בין 1 ל־8. הגרפיקה מופיעה שנית, אך הפעם שחור ולבן מתהפכים. כדי לצאת מאופן הפעולה – גרפיקה, עליך ללחוץ שנית על GRAPH, מכיוון שהמחשב לא יוצא ממנו בצורה אוטומטית.

עריכה בספקטרום

כאשר אתה נותן פקודות או מגיע לשלב כתיבת התכניות לספקטרום, תרצה בודאי לתקן טעויות בפקודות או שורות תכנית או לשנותן. ניתן לעשות זאת בקלות בעזרת תכנית העריכה.

כיצד לתקן טעות

אם אתה מנסה להכניס שורה או פקודה שגוייה בבייסיק, הספקטרום יציג סימן שאלה (?) מהבהב לפני השגיאה. כדי לתקן את הטעות לחץ על מקשי בקרת הסמן לכוון ימין או שמאל כדי להביא את הסמן לצד ימין של הטעות. כעת ניתן למחוק את הטעות ע״י הקשת DELETE או הוספת האות, מילת מפתח, מספר או סימן הנדרשים. אח״כ לחץ ENTER.

לדוגמה, נניח שאתה רוצה שהמחשב יבצע 7 כפול 8 ולא לחצת על SYMBOL SHIFT בעת הקשת סימן *. מה שהקשת בעצם הוא

PRINT 7b8

הספקטרום לא יכול לבצע פקודה זאת, ולכן בלחיצת ENTER יופיע סימן שאלה מהבהב לפני האות b, שם נמצאת השגיאה. כל שעליך לעשות הוא להביא את הסמן לימין הטעות וללחוץ DELETE כדי למחוק את ה־b.











אח״כ לחץ SYMBOL SHIFT ומקש B לקבלת הסימן *, בלווי ENTER כדי שהמחשב יציית לפקודה הנכונה. אינך חייב להזיז את הסמן לסוף השורה תחילה. הספקטרום מבצע את הפקודה ומציג את התוצאה.

כיצד לערוך שורת תכנית

כאשר אתה כותב תכנית, אתה בונה רצף של שורות פקודות ממוספרות הנקרא רישום. אם לאחר כתיבת התכנית, אתה 'רושם' זאת ע״י לחיצת K(LIST) ו־ENTER, תראה סימן < כנגד אחת משורות התכנית. אם לא, לחץ על מקש בקרת הסמן בכוון מעלה או מטה. אם תלחץ כעת על EDIT, שורה זאת תשוכפל בתחתית המסך וניתן יהיה לשנותה כמקודם בעזרת מקשי הסמן ו־DELETE. לחץ ENTER כדי להכניס את השורה החדשה לתכנית. אם ברצונך לערוך שורה אחרת, הזז את סימן < בעזרת מקש בקרת סמן מעלה או מטה עד לשורה הטעונה שינוי והקש EDIT. אם התהליך ארוך מדי, הקש LIST מלווה במספר השורה ואז הקש EDIT. בכל מקרה, השורה אותה דרשת תופיע בתחתית המסך וניתן יהיה לשנותה. כדי למחוק שורה שלמה מתכנית, פשוט הקש פנימה את מספר השורה בלבד ואח״כ לחץ ENTER. אם אתה מריץ תכנית המכילה שגיאה, תראה דוח מסך. דוחות אלה מוסברים בעמ' 74.

מחשבון הטלביזיה

הספקטרום +ZX יכול לבצע חישובים במהירות עצומה ובדיוק רב. כל מה שנדרש הוא כמה מספרים לעבוד עליהם וסימנים כמו + ו־ – האומרים לו מה לעשות עם המספרים. ראשית הקש פקודה זאת (את הסימן + תמצא על מקש K):

```
PRINT 6 + 2
```

זאת היא פקודה. כאשר אתה לוחץ ENTER, הפקודה נעלמת והתשובה, המספר 8, מופיע על המסך.

הספקטרום משתמש ב־5 סימנים הידועים כפעולות אריתמטיות עבור חישובים. תוכל לראות מה מבצע כל סימן בטבלה שלהלן. אתה יכול להשתמש בכולם באותה צורה יחד עם PRINT.

הקשת פקודות כמו PRINT 6 + 2 הופכת את הספקטרום למחשבון, אך הוא יכול לבצע דברים רבים שמחשבון רגיל אינו יכול לבצע. ראשית, הוא יכול להציג חישובים ואת תוצאותיהם יחד. הקש פקודה זו:

```
PRINT"6+2=";6+2
```

המחשב יגיב בהצגת:

```
6+2=8
```

מה שקורה הוא שפקודת PRINT גורמת לכל מה שנמצא בין מרכאות כפולות להיות מוצג על המסך, ולכן מופיע = 2 + 6. התווים בין סימני המרכאות מהווים מחרוזת. הסימן נקודה־פסיק מורה לספקטרום להציג את התוצאה מייד לאחר סימן השוויון.

טבלת סימני פעולות אריתמטיות

הסימנים שלהלן או 'פעולות אריתמטיות' משמשים בספקטרום לביצוע פעולות חשבוניות. שים לב לכך שהמחשב אינו משתמש בסימנים × או ÷.

סימן	מקש	פונקציה	דוגמה
+	K	חבר שני מספרים	8 + 2 = 10
–	J	חסר שני מספרים	8 – 2 = 6
*	B	הכפל שני מספרים	8 * 2 = 16
/	V	חלק שני מספרים	8 / 2 = 4
↑	H	העלה מספר ראשון בחזקת המספר השני	8 ↑ 2 = 64

התכנית הראשונה שלך

לאחר שפקודה התבצעה, הספקטרום שוכח אותה. אם אתה רוצה שהמחשב יחזור על החישוב, ניתן לכתוב אותו כתכנית. הקש פקודה זו ולחץ ENTER:

```
10 PRINT 6 + 2
```

הפעם, פקודה זו לא מתבצעת מייד. המחשב מציג במקום זאת את הפקודה על המסך. כעת הקש (R(RUN ו־ENTER. התוצאה 8 תופיע עכשיו.

כל הפקודה מהווה עתה תכנית מחשב. הצבת מספר בראש השורה גורמת לספקטרום למקם את הפקודה בזכרון, אך הוא אינו מבצע אותה אלא לאחר שהתבקש לעשות זאת. כאשר אתה מריץ תכנית ע"י הקשת (R(RUN ו־ENTER, הפקודה מתבצעת. ההוראה למחשב נקראת עתה הצהרה במקום פקודה, והיא מהווה שורה ממוספרת בתכנית. הצהרות תכנית מתבצעות תמיד לפי סדר מספרי השורות, ואלה עולות בדרך כלל בעשרות כך שניתן יהיה בעתיד להכניס שורות נוספות ביניהן במידת הצורך.

כעת, נגרום לספקטרום לעבוד קשה יותר. הקש את התכנית שלהלן. זכור ללחוץ ENTER לאחר הקשת כל שורה, וכאשר סיימת, לחץ (R(RUN ו־ENTER. לאחר שהרצת את התכנית, זה מה שאתה אמור לראות:

טבלת מספרים

```
10 LET n=1
20 PRINT n;"  ";
30 LET n=n+1
40 GO TO 20
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14
 15 16 17 18 19 20 21 22 23 24 2
5 26 27 28 29 30 31 32 33 34 35
36 37 38 39 40 41 42 43 44 45 46
 47 48 49 50 51 52 53 54 55 56 5
7 58 59 60 61 62 63 64 65 66 67
68 69 70 71 72 73 74 75 76 77 78
 79 80 81 82 83 84 85 86 87 88 8
9 90 91 92 93 94 95 96 97 98 99
100 101 102 103 104 105 106 107
108 109 110 111 112 113 114 115
116 117 118 119 120 121 122 123
124 125 126 127 128 129 130 131
132 133 134 135 136 137 138 139
140 141 142 143 144 145 146 147
148 149 150 151 152 153 154 155
156 157 158 159 160 161 162 163
164 165 166 167 168 169 170 171
172 173 174 175 176 177 178 179
180 181 182 183 184 185 186 187
188 189 190 191 192 193 194 195
196 197 198 199 200 201 202 203

scroll?
```

כל המספרים מ־1 עד 203 מוצגים על המסך. כעת לחץ על מקש כלשהו פרט למקשי N, רווח, STOP או BREAK. מערך חדש של מספרים יופיע על המסך.

תכנית זו משתמשת במשתנה. במקרה זה,

המשתנה נקרא n. ניתן להשתמש בכל אות או מילה – n מייצג כאן פשוט מספר. למשתנה ניתן ערך אשר משתנה תוך ריצת התכנית. בשורה 10, מילת המפתח LET משמשת לקביעת ערך המשתנה כ־1. שורה 20 מציגה את הערך כשהוא מלווה ברווח. בשורה 30 משתמשים שנית בפקודת LET, והפעם להגדיל את הערך ב־1, כך ש־n הופך ל־2. שורה 40 משתמשת במילת המפתח (יחידה) GOTO כדי לשלוח את התכנית חזרה לשורה 20, המציגה כעת 2. התהליך חוזר על עצמו שוב ושוב עד שהמספרים ממלאים את המסך.

כיצד לגרום לתכנית לבקש מספר

עצור את התכנית בלחיצת BREAK. כעת הקש פנימה שורה חדשה:

```
10 INPUT n
```

שורה זו מחליפה את שורה 10 הקודמת בתכנית. כאשר תריץ אותה, המחשב מחכה למספר שתכניס אליו. הקש פנימה מספר כלשהו ולחץ ENTER. כעת המספרים יתחילו במספר שהכנסת. הסיבה לכך היא שפקודת INPUT n קובעת את ערך n כשווה למספר שהכנסת. INPUT מורה למחשב לקלוט נתון במהלך ריצת תכנית.

תכנות טבלת כפל

לחץ על כפתור RESET למחיקת התכנית הישנה והקש את התכנית הבאה. תכנית זו גורמת לספקטרום להכפיל מספרים. הקש מספר כלשהו ועל המסך תופיע טבלת כפל עבור מספר זה. לחץ על מקש כלשהו פרט למקשי N, BREAK או רווח, והטבלה תמשיך. לחץ BREAK והרץ את התכנית שנית ליצירת טבלה חדשה. להלן מופיעה התכנית ומה שתוכל לראות אם תקיש פנימה 3 ואח"כ 146:

טבלת כפל

```
10 LET x=1
20 INPUT n
30 PRINT n;"*";x;"=";n*x
40 LET x=x+1
50 GO TO 30
```

```
3*1=3
3*2=6
3*3=9
3*4=12
3*5=15
3*6=18
3*7=21
3*8=24
3*9=27
3*10=30
3*11=33
3*12=36
3*13=39
3*14=42
3*15=45
3*16=48
3*17=51
3*18=54
3*19=57
3*20=60
3*21=63
3*22=66

scroll?
```

```
146*1=146
146*2=292
146*3=438
146*4=584
146*5=730
146*6=876
146*7=1022
146*8=1168
146*9=1314
146*10=1460
146*11=1606
146*12=1752
146*13=1898
146*14=2044
146*15=2190
146*16=2336
146*17=2482
146*18=2628
146*19=2774
146*20=2920
146*21=3066
146*22=3212

scroll?
```

מדוע עליך להשתמש בסוגריים

לעיתים תצטרך להשתמש בסוגריים במהלך החישוב. הקש את שתי הפקודות הבאות והשווה את התוצאות:

```
PRINT 6+2/4
PRINT (6+2)/4
```

הראשונה נותנת 6.5 ואילו השנייה נותנת 2. הסיבה לתוצאות שונות אלה היא בכך שלמחשב יש מערכת עדיפויות פנימית לצורך חישובים. המחשב מבצע ראשית ↑, אח"כ * או /, ולבסוף + או –, אך פעולות בסוגריים מתבצעות תמיד ראשונות. ולכן, בפקודה הראשונה שלעיל, המחשב מחלק ראשית 2 ב־4 ואחר כך מחבר את התוצאה (0.5) ל־6. בפקודה השנייה, המחשב מחבר 6 ו־2, ואח"כ מחלק את התוצאה ב־4.

ביצוע סימני פיסוק

הספקטרום משתמש במבחר של סימני פיסוק. סימנים אלה חשובים מאחר ורבים מהם משמשים גם כהוראות למחשב, ומשפיעים על הדרך בה הוא מבין שורת תכנית או יוצר תצוגה.

- ; **נקודה פסיק** – כאשר משתמשים בו עם PRINT, המחשב מציג את שני הפרטים משני צידיו צמודים אחד לשני על המסך.
- : **נקודתיים** – מסמן סיום הצהרה אחת בשורת תכנית ואת תחילתה של ההצהרה הבאה.
- " **סימן מרכאות** – כל התווים הנמצאים בין מרכאות אינם נחשבים כמספרים או משתנים אלא כטקסט. מרכאות פותחות וסוגרות מחרוזת.
- , **פסיק** – כאשר מופיע בפקודת PRINT, הוא אומר למחשב להציג את הפרט הבא באמצע השורה או בתחילת השורה הבאה. אל תשתמש בו לציון אלפים או מליונים.
- . **נקודה** – נקודה עשרונית או סיום משפט.

הצבע וכיצד להשתמש בו

הספקטרום +ZX יכול ליצור 8 צבעים שונים, כאשר לכל צבע יש מספר קוד. ניתן להשתמש בכל צבע ב־3 דרכים שונות – כצבע גבול מסך, כצבע דיו, וכצבע נייר (רקע פנימי במסך).

קודי צבע בספקטרום +ZX

טבלה זאת מתארת את הצבעים והקודים הקיימים בספקטרום. אינך חייב לזכור קודים אלה; מקשי המספרים היוצרים אותם מסומנים גם בשמות הצבעים. (שמות אלה אינם מילות מפתח).

צבע	מספר
שחור	0
כחול	1
אדום	2
סגול	3
ירוק	4
תכלת	5
צהוב	6
לבן	7

הגוונים שתקבל למעשה על מקלט הטלביזיה תלויים במקלט עצמו ובכוונון עצמת הצבע, ופקדי ניגודיות (קונטרסט) ובהירות. זכור שעליך להשתמש במקלט טלביזיה צבעוני.

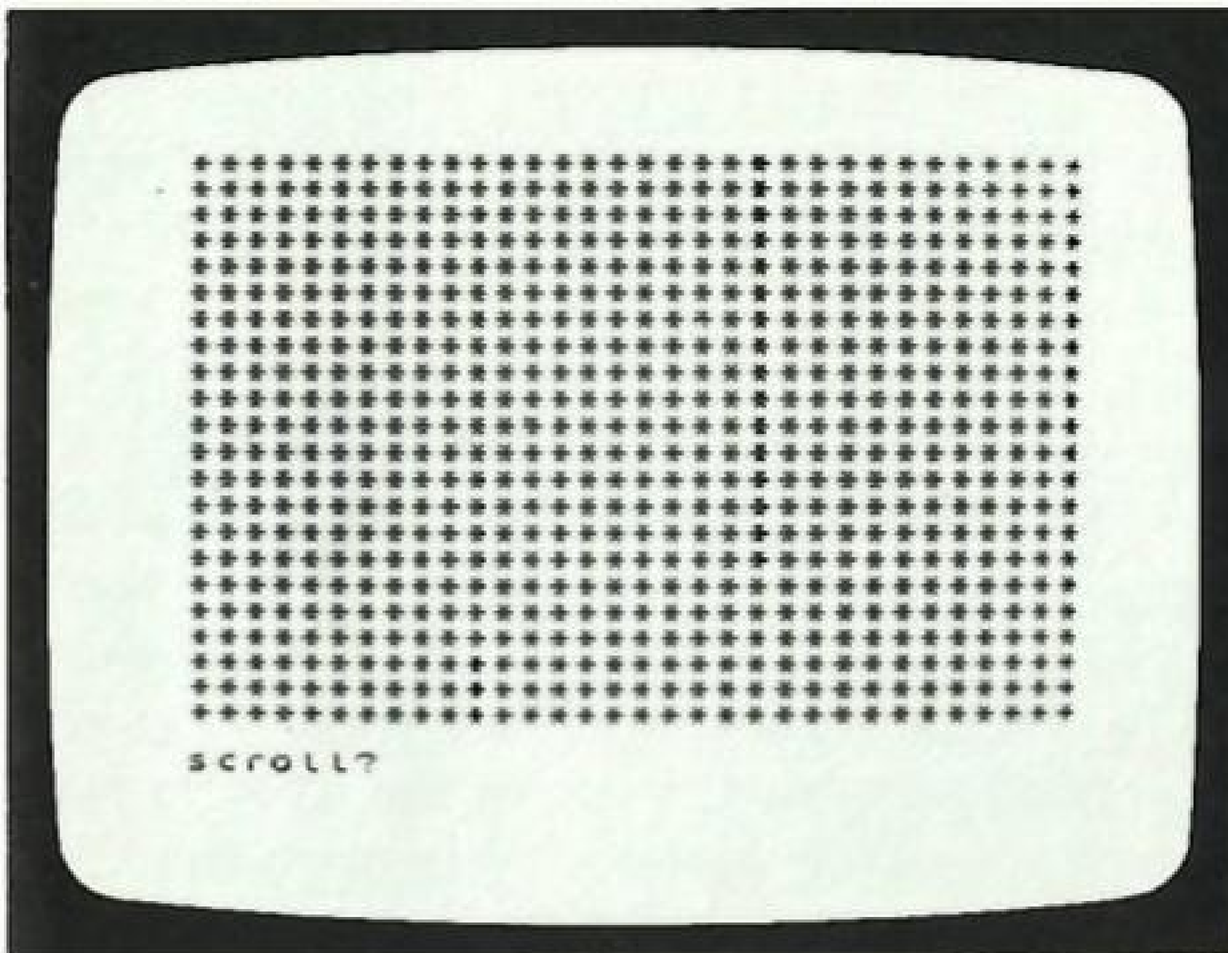
שלושת אופני השימוש בצבע בספקטרום

ניתן לבקר את הצבע בשלש דרכים שונות. צבע הגבול הוא צבע גבולות המסך מסביב לשטח התצוגה העקרי. צבע הדיו הוא הצבע בו מופיעים תווים (אותיות, מספרים, סימנים וצורות גרפיים) ונקודות או קוים. צבע הנייר הוא צבע הרקע, או על פני כל שטח התצוגה או בריבוע סביב כל תו.

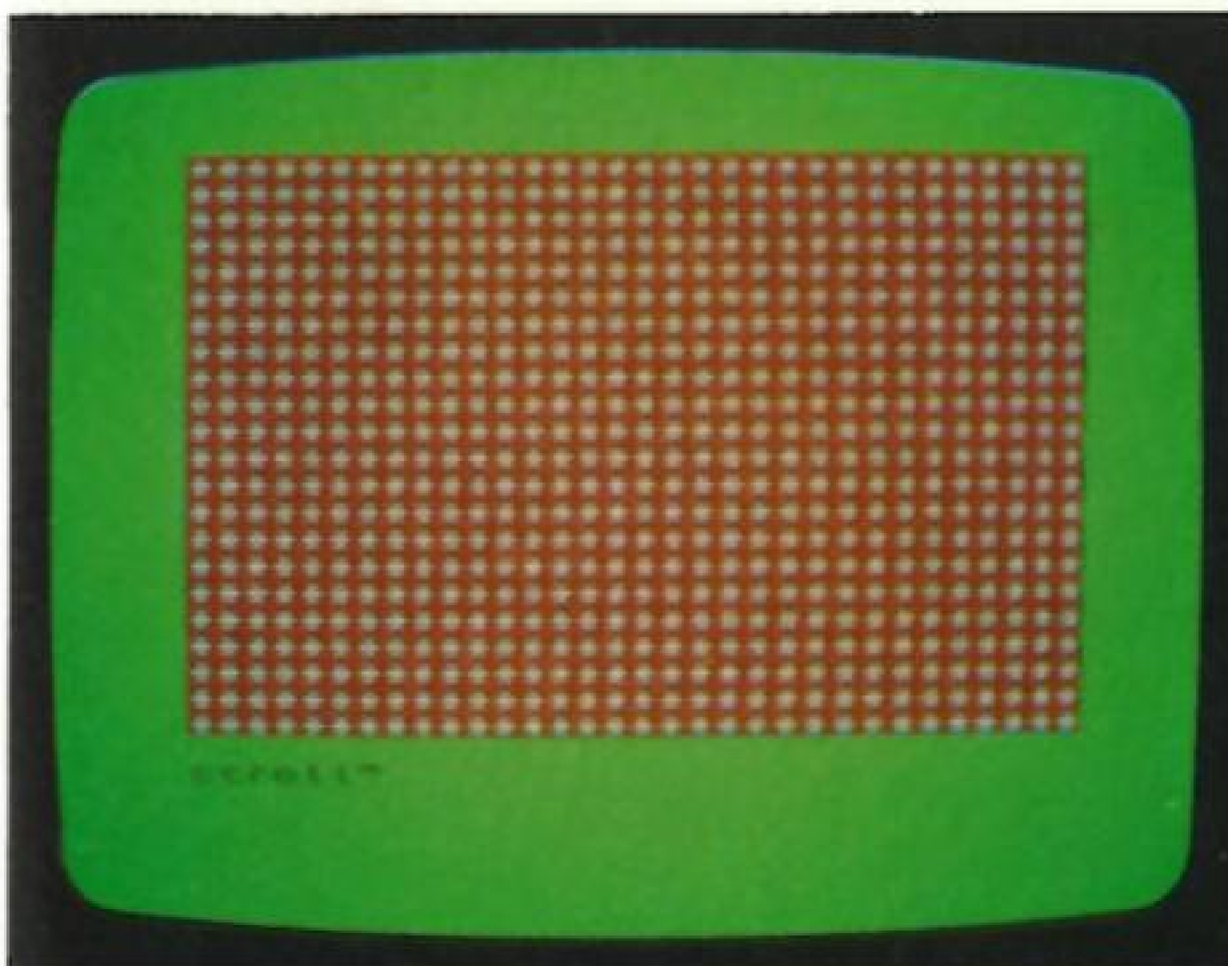
כאשר אתה מפעיל את הספקטרום, הוא משתמש בצבעים הקבועים מראש. צבע הדיו שחור, וצבעי הגבול והמסך לבנים. אתה יכול לשנות צבעים אלה כהרף עיין במתן פקודות ישירות מהמקלדת. נתקלת כבר באפשרות זאת בעמ' 6-7, בהם השתמשת בפקודת BORDER לבדיקת החיבור הנכון של הטלביזיה והספקטרום מבחינת הצבע. כעת לחץ על כפתור RESET והקש והרץ תכנית פשוטה זאת:

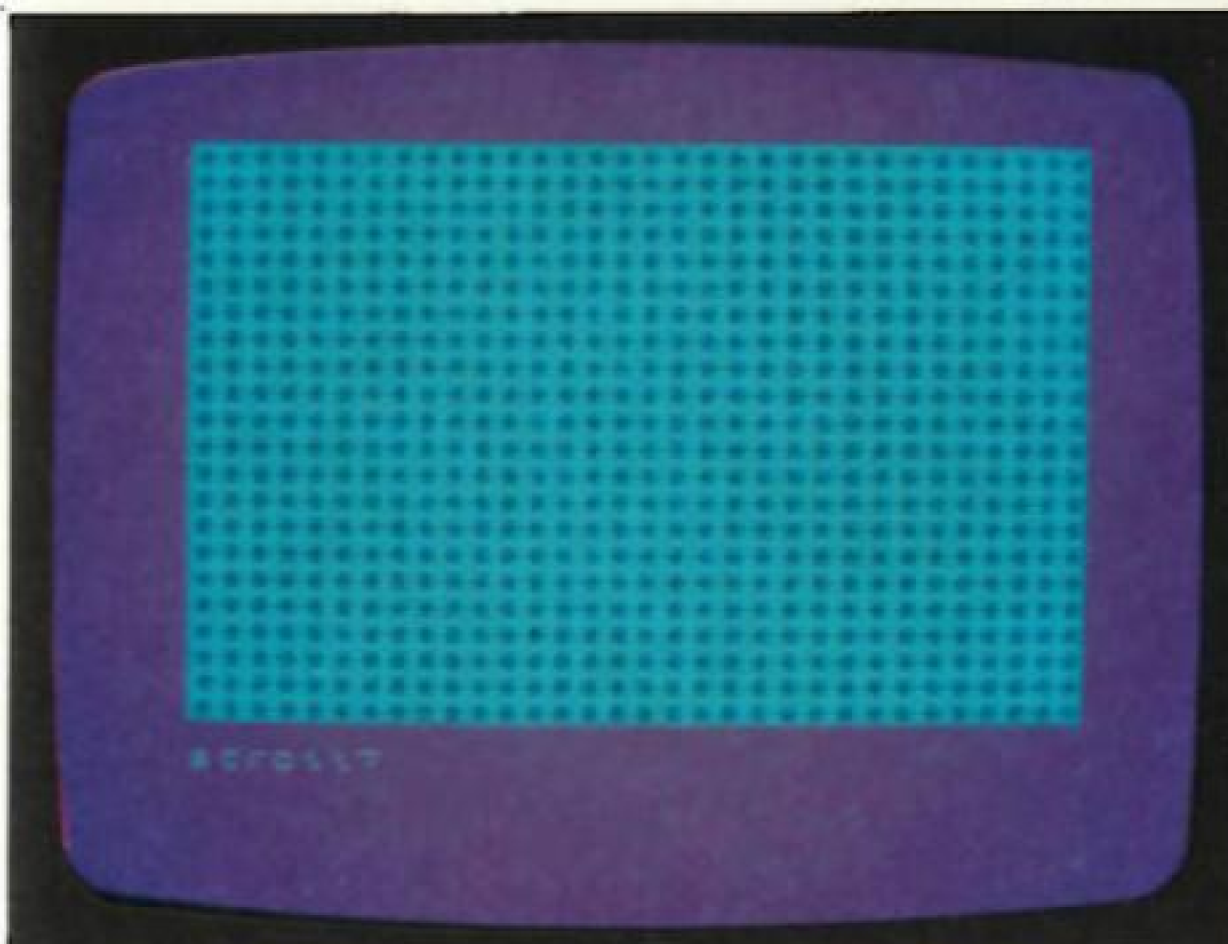
בדיקת צבע

```
10 PRINT "*";
20 GO TO 10
```



שורות של כוכבים נבנות בשחור ולבן. כעת לחץ BREAK והקש פנימה כמה פקודות צבע. הקש פנימה את מילות המפתח BORDER, INK, ו־PAPER כשהן מלוות במספר בתחום 0 עד 7, לחץ ENTER אחרי כל פקודה, והרץ שנית את התכנית. להלן מופיעות שתי תצוגות, הראשונה עם BORDER 4, PAPER 2 ו־INK 7, והשנייה עם BORDER 3, PAPER 5 ו־INK 1.





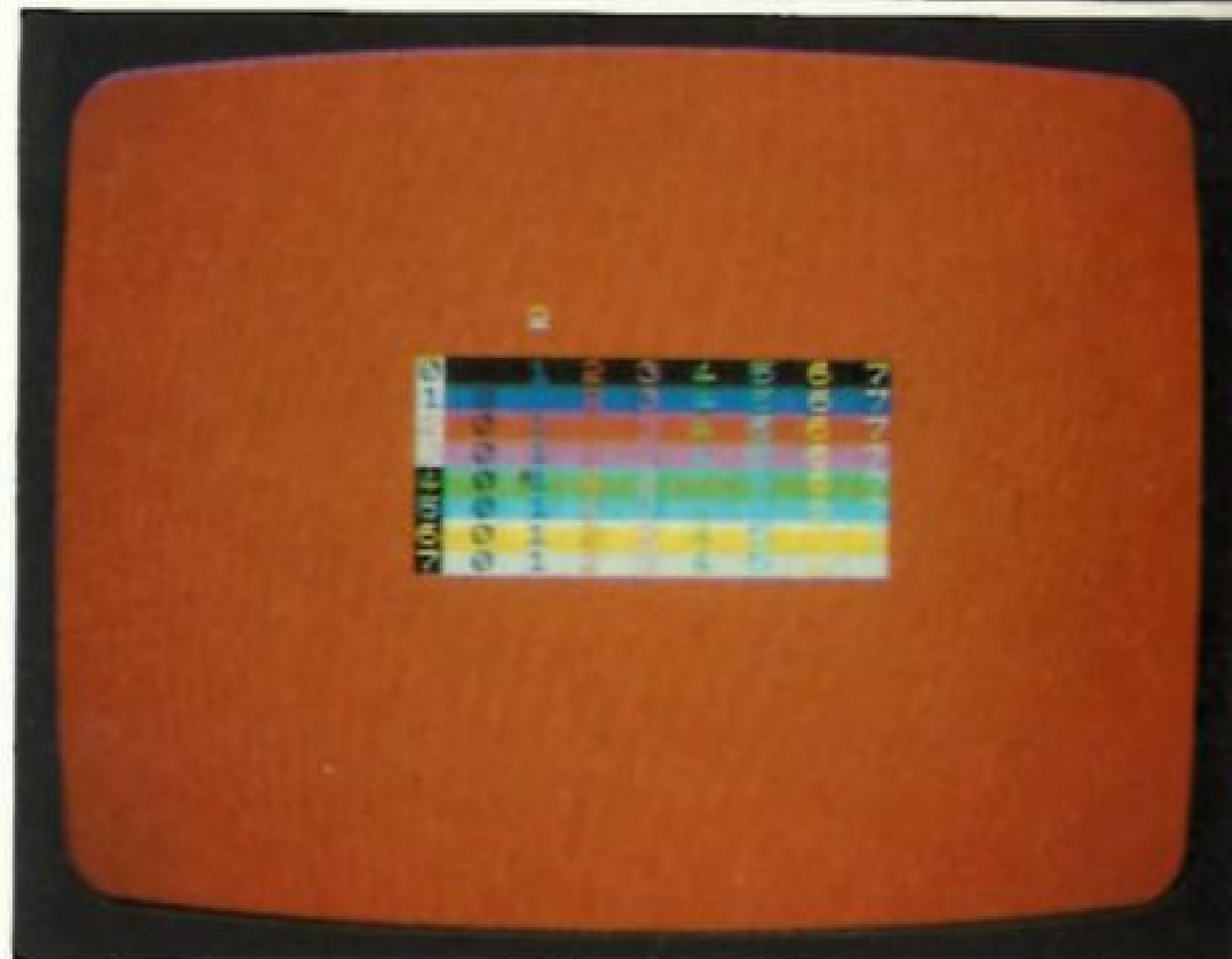
כיצד לכתוב תכניות עם צבע

ניתן להשתמש במילות המפתח PAPER ,BORDER ו־INK בתוך תכנית כדי שטקסט, טבלאות, צורות ותמונות יופיעו בכל מיני צבעים. השימוש ב־BORDER בשורת תכנית גורם לשינוי צבע גבולות המסך ברגע שהספקטרום הגיע לשורה זו. כאשר INK מופיע בשורה משלו, הוא נותן צבע דיו חדש לכל התווים או השורות שיופיעו על המסך. פקודת PAPER בשורה משלה משנה את צבע הנייר אך רק סביב התווים (כולל גם נקודות או שורות). אם אתה רוצה לשנות את צבע כל הרקע, עליך להקיש PAPER בלווי CLS.

ניתן גם להשתמש ב־INK ו־PAPER אחרי PRINT. במקרה זה, רק אותם תווים המוצגים ע"י PRINT, יקבלו את הצבעים מהוראות INK ו־PAPER. התכנית הבאה מדגימה את כל צבעי הגבול, דיו ונייר. כמו כן היא מראה כיצד להשתמש ב־INK ו־PAPER לאחר PRINT.

צירופי צבע

```
 10 FOR b=0 TO 7
 20 BORDER b: PAPER b: CLS
 30 PRINT AT 6,12; INK 9;b
 40 FOR p=0 TO 7
 50 PRINT AT p+8,8; INK p; PAPE
R 9;p;
 60 BEEP 0.5,b*p-20+p
 70 FOR i=0 TO 7
 80 PRINT INK i; PAPER p;" ";i;
 90 BEEP 0.01,i*5
100 NEXT i
110 NEXT p
120 NEXT b
```



כאשר תריץ תכנית זאת, תראה את כל הצירופים האפשריים של גבול, נייר ודיו. בתכנית יש 3 משתנים, b עבור מספר גבול, i עבור מספר הדיו ו־p עבור מספר נייר. BEEP יוצר את הקול, והשורות המתחילות ב־FOR ו־NEXT מציינות התחלת וסוף לולאת תכנית אשר משנה לפי הסדר את כל מספרי הצבעים מ־0 עד 7. על השימוש ב־FOR ו־NEXT בלולאות תכנית תמצא יותר בעמ' 27. שים לב לכך ש־INK ו־PAPER יכולים לקבל את הערך 9. כך הופכים את צבעי הדיו או הנייר לשחור או לבן כך שיובלטו כנגד הרקע או התו.

תכנות דיאגרמות עמודות צבעוניות

התכנית הבאה משתמשת בצבעי הספקטרום ליצירת דיאגרמת עמודות. הדיאגרמה מתארת 12 טמפרטורות יומיות בעמודות צהובות עם מספרים. בשורה 60, הכנס שני רווחים בין סימני המרכאות.

דיאגרמת עמודות

```
 10 BORDER 0: PAPER 1: CLS
 20 LET c=4
 30 FOR x=1 TO 12
 40 READ t
 50 FOR l=21 TO 21-t STEP -1
 60 PRINT PAPER 6;AT l,c;"  "
 70 NEXT l
 80 PRINT INK 2;AT 20-t,c;t
 90 LET c=c+2
100 NEXT x
110 DATA 20,15,13,16,19,20,18,1
1,12,19,14,17
```



כעת הוסף את השורות הבאות, והקש פנימה את שורה 110 כמתואר. הדיאגרמה תופיע עתה בשני צבעים. כדי ללמוד יותר על פקודות READ ו־DATA, עיין בעמ' 33.

דיאגרמת עמודות כפולה

```
 85 READ t
 86 FOR l=21 TO 21-t STEP -1
 87 PRINT PAPER 3;AT l,c;"  "
 88 NEXT l
 89 PRINT INK 1; PAPER 5;AT 20-
t,c;t
110 DATA 20,6,15,4,13,5,16,6,19
,10,20,8,18,6,11,4,14,6,19,8,14,
9,17,7
```



גרפיקת "עשה זאת בעצמך"

הספקטרום +ZX יכול לתת גרפיקה בהפרדה גבוהה ובהפרדה נמוכה. שני הסוגים יכולים להופיע יחד על המסך. תצוגות גרפיקה בהפרדה נמוכה בנויות מגושי צבע. בשני העמודים שלהלן, תלמד כיצד ליצור גושים אלה במקלדת, וכיצד למקם אותם על המסך.

המסך בהפרדה נמוכה

במסך בהפרדה נמוכה יש 32 מקומות בהם ניתן למקם תוים לרוחב המסך ו־22 מקומות לאורך המסך. לכל נקודה במסך יש זוג מספרים המזהה אותה. ראשית בא מספר השורה, שהוא מספר השורות בו יש לרדת לאורך המסך כדי להגיע לנקודה. השורה העליונה היא שורה 0 והשורה התחתונה היא שורה 21. אחר כך מגיע מספר העמודות לרוחב המסך אותו יש לעבור כדי להגיע לנקודה. העמודה השמאלית היא עמודה 0 והעמודה הימנית היא עמודה 31. (בעמ' 80 תוכל לראות את מערך רשת המסך בהפרדה נמוכה). התכנית הבאה ממלאת מקומות תוים אלה בצבעים. מילת המפתח RND (על מקש R) בוחרת צבע דיו אקראי.

רבועים אקראיים

```
10 BORDER 1: INK RND*7
20 PRINT "█";
30 GO TO 10
```



כאן מופיעים רבועים על פני כל המסך. כדי לגרום להופעת תו בנקודה מסויימת, עליך להשתמש במילת המפתח AT יחד עם PRINT. AT באה אחרי PRINT ומלווה במספר השורה, פסיק, מספר העמודה ונקודה־פסיק. הפקודה הבאה:

```
PRINT AT 11,16;"*"
```

לדוגמה, מציגה כוכבית בשורה 11, עמודה 16, שם נמצא מרכז המסך.

כיצד לצייר צורות קשת בענן

דרך טובה ליצירת צורות צבעוניות היא השימוש בלולאות FOR NEXT בתכניות הגרפיקה. לולאות FOR NEXT הן חלקים בתכנית החוזרים על עצמם מספר מסויים של פעמים. בשורה הפותחת לולאה, אתה יכול להורות למחשב כמה פעמים יש לבצע את הלולאה. כאשר המחשב מבצע זאת, ניתן להשתמש בכך, לדוגמה, להצבת תוים על המסך.

כיצד לבחור תווים גרפיים

לספקטרום + ZX יש מערך תוים גרפיים במקלדת המאפשרים תכנות קל של גרפיקה בהפרדה נמוכה. ניתן לראותם על מקשים 1 עד 8.
כדי ליצור תוים גרפיים על המסך, לחץ על מקש GRAPH ולחץ על מקשים 1 עד 8, תוך שימוש במקש הרווחים בין אחד לשני. התוים הגרפיים מופיעים בתחתית המסך. החלק הלבן של כל תו במקש הוא צבע הדיו, והחלק השחור צבע הנייר. כעת לחץ על המקשים שנית, יחד עם לחיצת CAPS SHIFT באותו זמן. הפעם יופיעו התוים כאשר צבעי הדיו והנייר הפוכים.
זוהי הדרך להציב תווים גרפיים בשורות תכנית. כדי לצאת מבחירת גרפיקה ולהחזיר את מקשי המספרים למצבם הרגיל, פשוט לחץ שנית על מקש GRAPH.

Key 8
משתמשים במקש זה לעתים קרובות עם GRAPH ו־CAPS SHIFT כדי ליצור ריבוע מלא של צבע.

GRAPH
משתמשים במקש זה כדי להעביר את הספקטרום לאופן פעולה — גרפיקה.



TRUE VIDEO	INV VIDEO	BLUE DEF FN ! 1	RED FN @ 2	MGNTA LINE # 3	GREEN OPEN# $ 4	CYAN CLOSE# % 5	YELLW MOVE & 6	WHITE ERASE ' 7	POINT (8

אתה אינך מוגבל לתכנות לולאה אחת בלבד בכל פעם. ניתן להציב לולאה אחת בתוך אחרת, ולרוב עם תוצאות שימושיות מאד. התכנית הבאה תראה לך כיצד שתי לולאות FOR NEXT (אחת 'מקוננת' בתוך האחרת) יכולות לשמש עבור החלפת הצבעים והמקומות שנוצרו ע"י INK ו־AT. אתה יכול לראות כיצד לתכנת לולאות אלה בפיסקה שבתחתית עמוד זה.

קשת בענן

```
 5 BORDER 0: PAPER 5: CLS
10 LET x=1
20 FOR l=0 TO 21
30 FOR c=1 TO 6
40 PRINT INK c;AT l,c+x;"█"
50 NEXT c
60 LET x=x+1
70 NEXT l
```



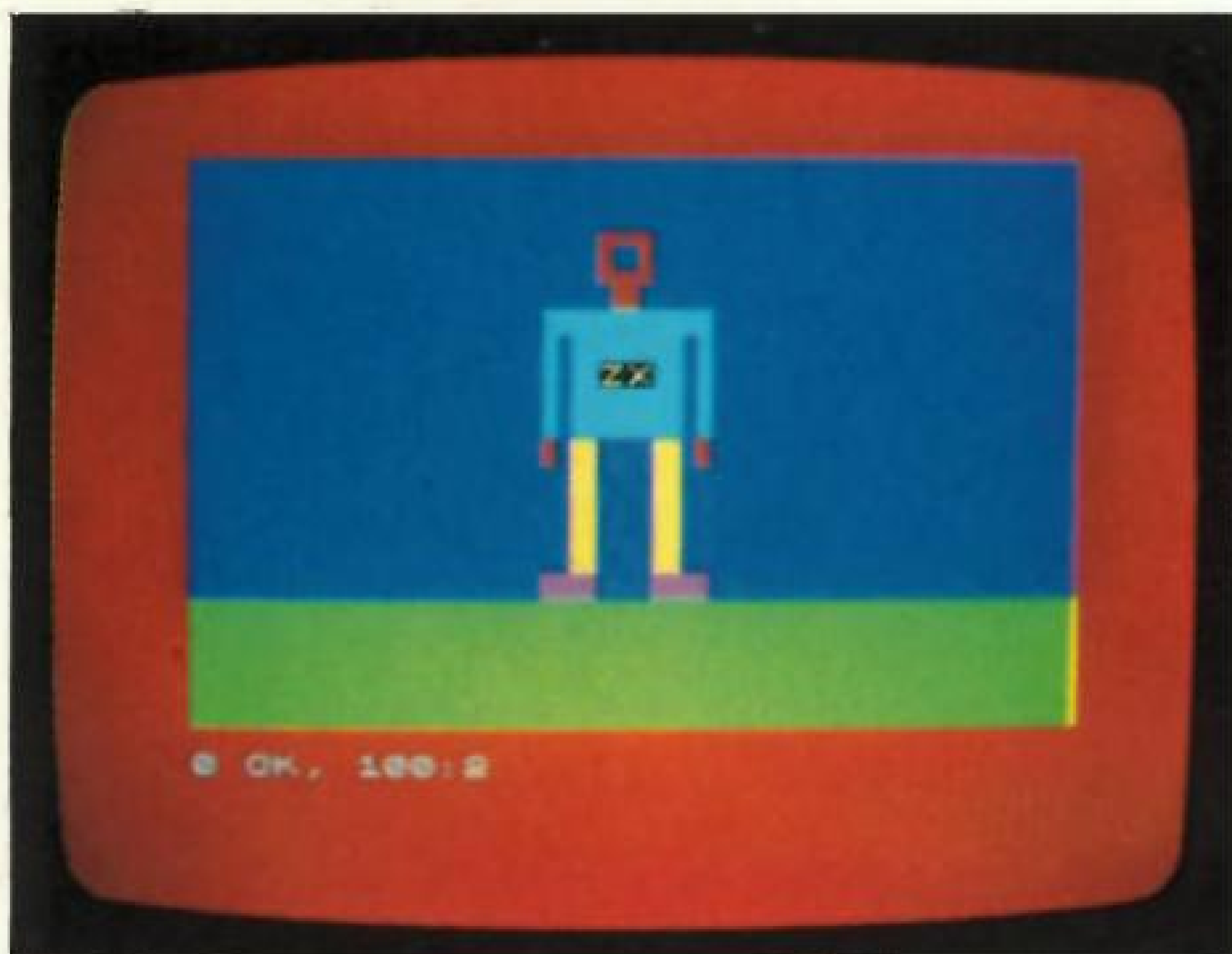
תכנות תמונות

בגרפיקה בהפרדה נמוכה, אתה יכול 'לצייר' תמונות בעזרת חישוב המקומות והצבעים של התוים הגרפיים. ניתן לתכנן את התמונה תוך שימוש ברשת הנקודות בהפרדה נמוכה בעמ' 80. ואז, תוך בחירת התוים הגרפיים בדרך המתוארת בעמוד הבא, הקש פנימה את שורות התכנית אחת אחרי השנייה ליצירת התמונה.

התכנית הבאה מראה את סוג התוצאות שניתן לקבל. את כל הצורות המופיעות אפשר למצוא על מקשי המספרים. אתה יכול לחכות עד הקשת כל המספרים לפני הרצת התכנית, אך אם, במקום זאת, תריץ את התכנית בתום הקשת כל שורה, תוכל לראות כיצד חלקי הרובוט השונים מורכבים יחד. (זכור שאם הקשת תוים גרפיים שגויים, ניתן לערוך אותם בדיוק כמו עריכת מספרים או אותיות שגויות.)

רובוט ZX

```
  5 BORDER 2: PAPER 1: CLS
 10 PRINT INK 2;AT 3,15;"█▀█"
 15 PRINT INK 2;AT 4,15;"█▄█"
 20 PRINT INK 2;AT 5,15;" █ "
 30 PRINT INK 5;AT 6,13;"███████"
 40 FOR l=7 TO 10: PRINT INK 5;AT l,13;"▌█████▐": NEXT l
 45 PRINT INK 6; PAPER 0;AT 8,15;"ZX"
 50 PRINT INK 2;AT 11,13;"▌    ▌"
 60 FOR l=11 TO 15: PRINT INK 6;AT l,14;"█  █": NEXT l
 70 PRINT INK 3;AT 16,13;"██";TAB 17;"██"
 80 FOR l=17 TO 21: FOR c=0 TO 31
 90 PRINT INK 4;AT l,c;"█"
100 NEXT c: NEXT l
```



מילת המפתח TAB המופיעה אחרי PRINT בשורה 70 משמשת למיקום תו לאורך השורה בה עובד המחשב באותו זמן. הקשת TAB המלווה במספר בין 0 ו־31, לקביעת מיקום עמודה.

כיצד להשתמש בלולאות FOR NEXT

לולאת FOR NEXT מתחילה תמיד בשורה המכילה את מילות המפתח FOR ו־TO, יחד עם משתנה וערכיו ההתחלתי והסופי. לדוגמה:

```
30 FOR C=1 TO 6
```

כאן המשתנה הוא C. הלולאה אותה מתחילים תכיל שורה(ות) הגורמת למחשב לחזור על הפעולה. שורות התכנית יכולות אף הן להשתמש במשתנה C. לולאות FOR NEXT מסתיימות תמיד במילת המפתח NEXT והמשתנה. לדוגמה:

```
50 NEXT C
```

כאשר התכנית רצה, כל הלולאה מ־FOR ועד NEXT חוזרת על עצמה מספר פעמים קבוע מראש. המשתנה מתחיל בערך הראשון לפני TO וגדל ב־1 בכל פעם עד שיגיע לגבול אחרי TO. במקרה זה, הלולאה חוזרת 6 פעמים, כאשר C מתחיל בערך 1 והופך להיות 2, 3, 4, 5 ולבסוף 6.

בתכנית הראשונה בעמ' 25, מופיעות 3 לולאות בתוך 'קן'. זה אומר שעבור כל מחזור לולאה 'חיצונית', הלולאה ה'מרכזית' תעבור בכל המחזורים. הלולאה ה'פנימית' מבצעת את מחזוריה בתכיפות רבה, בכל פעם שהלולאה ה'מרכזית' משלימה סיבוב פעולה אחד.

לוח טיוטה על המסך

הגרפיקה בספקטרום +ZX אינה מוגבלת לצורות מגושמות בהפרדה נמוכה. בעזרת יכולת ההפרדה הגבוהה ניתן להשתמש בספקטרום ליצירת תמונות מפורטות עם תוים חדים, קווים ישרים או עקומים וזוויות.

גרפיקה בהפרדה גבוהה בנוייה מנקודות רבות הממוקמות בזו אחר זו ליצירת קו או מילוי צורה בצבע. כל נקודה מהווה 1/64 מגודל הריבוע של גרפיקה בהפרדה נמוכה. אם תקיש את הפקודה:

PLOT 128 ,87

כעת תראה נקודה כזאת במרכז המסך.

הנקודות המשמשות בגרפיקה בהפרדה גבוהה נקראות פיקסלים, בקיצור מאנגלית ל"תאי תמונה". בדומה לתו בהפרדה נמוכה, כל תא תמונה דורש שני מספרים להגדרת מיקומו. אלה אינם אותם המספרים המשמשים בתצוגות בהפרדה נמוכה.

רשת הנקודות בהפרדה גבוהה

רשת הנקודות בהפרדה גבוהה בנוייה מ־256 תאי תמונה לרוחב המסך ו־176 לאורכו. ברם, שלא כמו בתצוגות בהפרדה נמוכה, המספר הראשון הוא הקואורדינטה האפקית – המיקום לרוחב המסך. מספרי נקודות אלה נעים מ־0 בפינה השמאלית עד 255 בפינה הימנית. המספר השני הוא הקואורדינטה האנכית, אך כאן המספרים נעים בין 0 בתחתית ל־175 בראש המסך. נקודה 0,0 היא הפינה השמאלית התחתונה במסך, ולא הפינה השמאלית העליונה כמו בהפרדה נמוכה. ראה טבלת רשת נקודות בהפרדה גבוהה בעמ' 80.

התוייה וציור

אתה זקוק לשלש מילות מפתח בלבד ליצירת גרפיקה בהפרדה גבוהה – PLOT, DRAW ו־CIRCLE. PLOT מלווה בקואורדינטות האפקיות והאנכיות כשהן מופרדות בפסיק, ואז ממוקם תא תמונה בנקודה זאת. DRAW מלווה אף היא בשני מספרים המופרדים בפסיק, אך אלה אינם קואורדינטות הנקודה. במקום זאת, אלה הם המרחקים בפיקסלים מנקודה אחת לנקודה אחרת לאורך ולרוחב המסך, ואז DRAW מציירת קו בין שתי הנקודות.

הנקודה ההתחלתית היא 0,0 אם לא השתמשנו מקודם ב־PLOT או DRAW בתכנית. אם השתמשנו בהן מקודם, אזי נקודה זו היא נקודת PLOT האחרונה, או הנקודה אליה הגיעו בפקודת DRAW האחרונה, מה שבא אחרון. הצהרת DRAW תצייר עתה את הקו לנקודה החדשה. אם הקו אמור לפנות לשמאל או לתחתית המסך, אזי המרחקים האפקיים והאנכיים מקבלים ערכים שליליים (מינוס). נסה תכנית זאת:

כוכב

```
10 INK 2
20 PLOT 128,174
30 DRAW 70,-140
40 DRAW -152,80
50 DRAW 164,0
60 DRAW -150,-80
70 DRAW 70,140
```

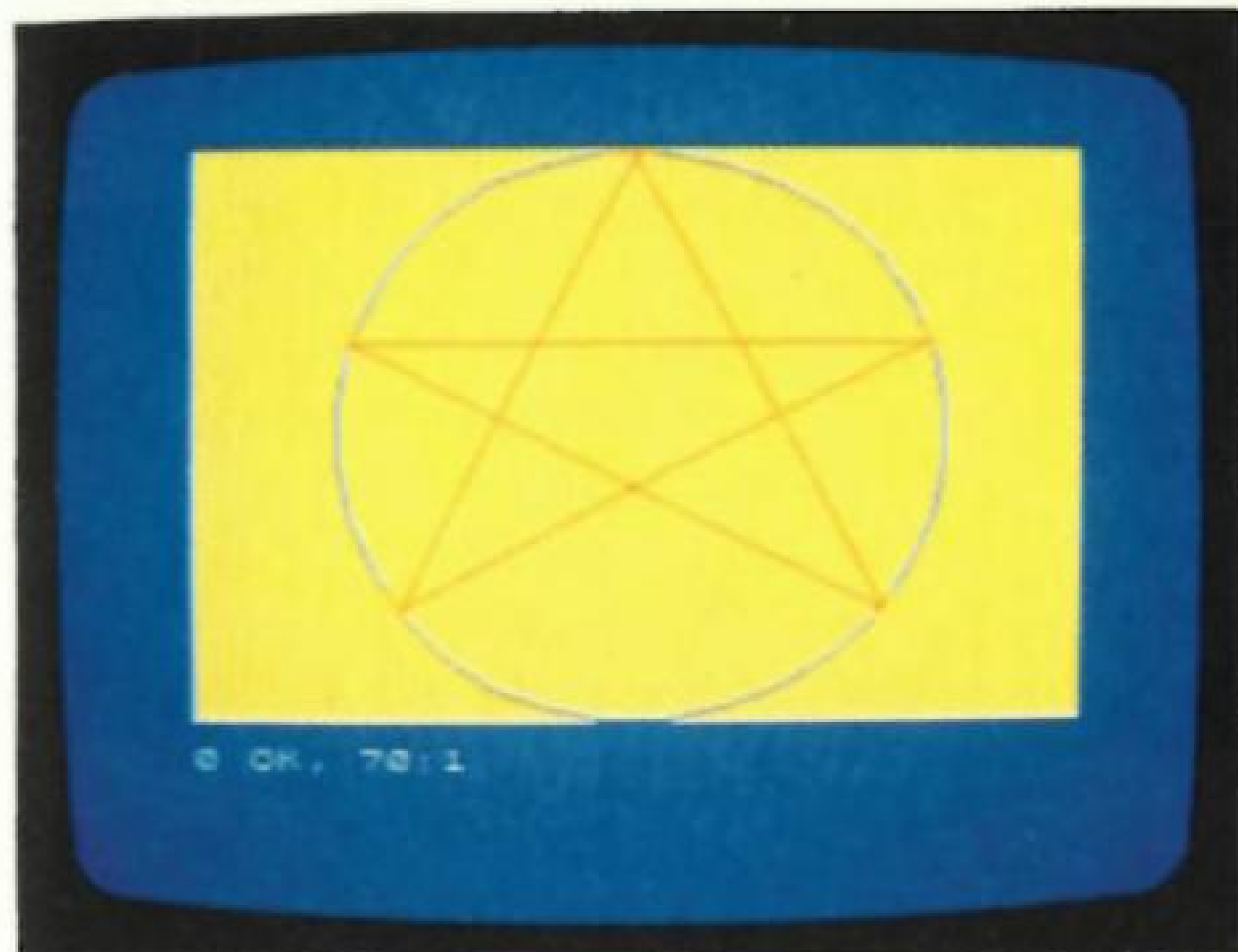


PLOT מזיזה את נקודת ההתחלה אל ראש המסך. אחר כך מציירים את 5 הקוים האדומים בעזרת 5 הצהרות DRAW.

כעת הוסף לתכנית את השורות הבאות:

```
4 BORDER 1:PAPER 6:INK 1:CLS
5 CIRCLE 128,87,87
```

הרץ שוב את התכנית, והכוכב האדום יופיע בתוך



עיגול על רקע נייר צבעוני.

CIRCLE דורשת 3 ערכים. שני הראשונים נותנים את מיקום מרכז העיגול, והמספר השלישי נותן את הרדיוס. אתה יכול להוסיף ערך שלישי גם להצהרות

DRAW. נסה ערכים בין 2 ר־2- עם התכנית וראה מה קורה.

כיצד למלא צורות

אתה יכול ליצור בקלות צורות מלאות בהפרדה גבוהה ע"י ציור קוים רבים כשהם קרובים אחד לשני. הדבר ניתן להיעשות בלולאת FOR NEXT אשר משנה את נקודות המיקום ב־DRAW כך שהן עולות ב־1 בכל פעם.

משולש מלא

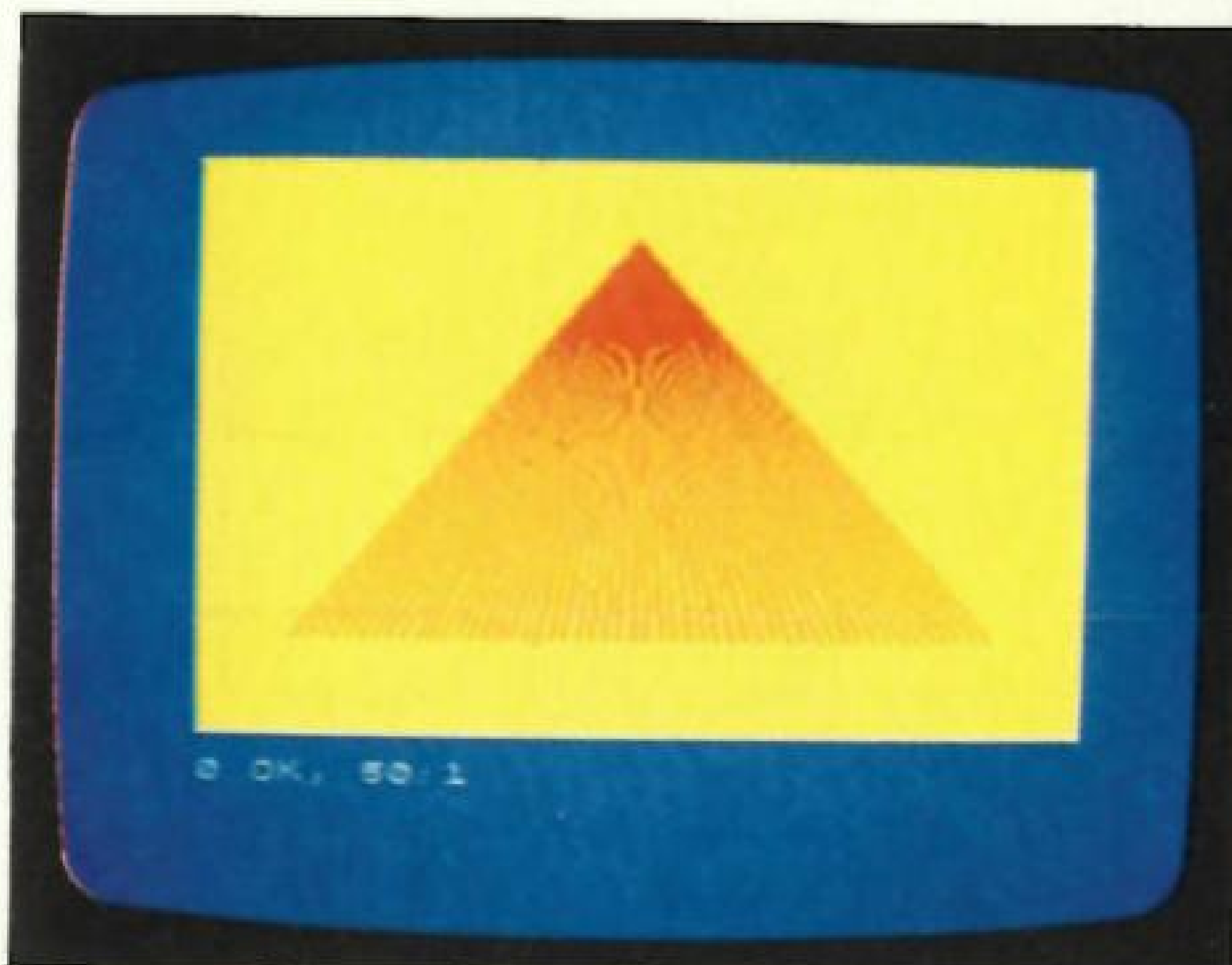
```
 10 BORDER 1: PAPER 6: INK 2: C
LS
 20 FOR x=-100 TO 100
 30 PLOT 128,150
 40 DRAW x,-120
 50 NEXT x
```



אתה יכול לקבל אפקט מעניין אם תצייר קוים מרווחים מעט זה מזה. ניתן לעשות זאת בהוספת מילת המפתח STEP ומספר להצהרת FOR. בטכניקה זו משתמשים בתכנית "הזריחה המנצנצת" בעמ' 11, אך היא תפעל על תכנית המשולש באותה צורה. הקש שורה 20 אחרת והרץ שנית את התכנית.

```
20 FOR X=-100 TO 100 STEP 4
```

הפעם תופיע הצורה דמויית־מניפה המופיעה להלן. הסיבה לכך היא ש־STEP גורמת ל־x לעלות בקפיצות של 4 במקום להגדילו ב־1 בכל פעם שמציירים קו.



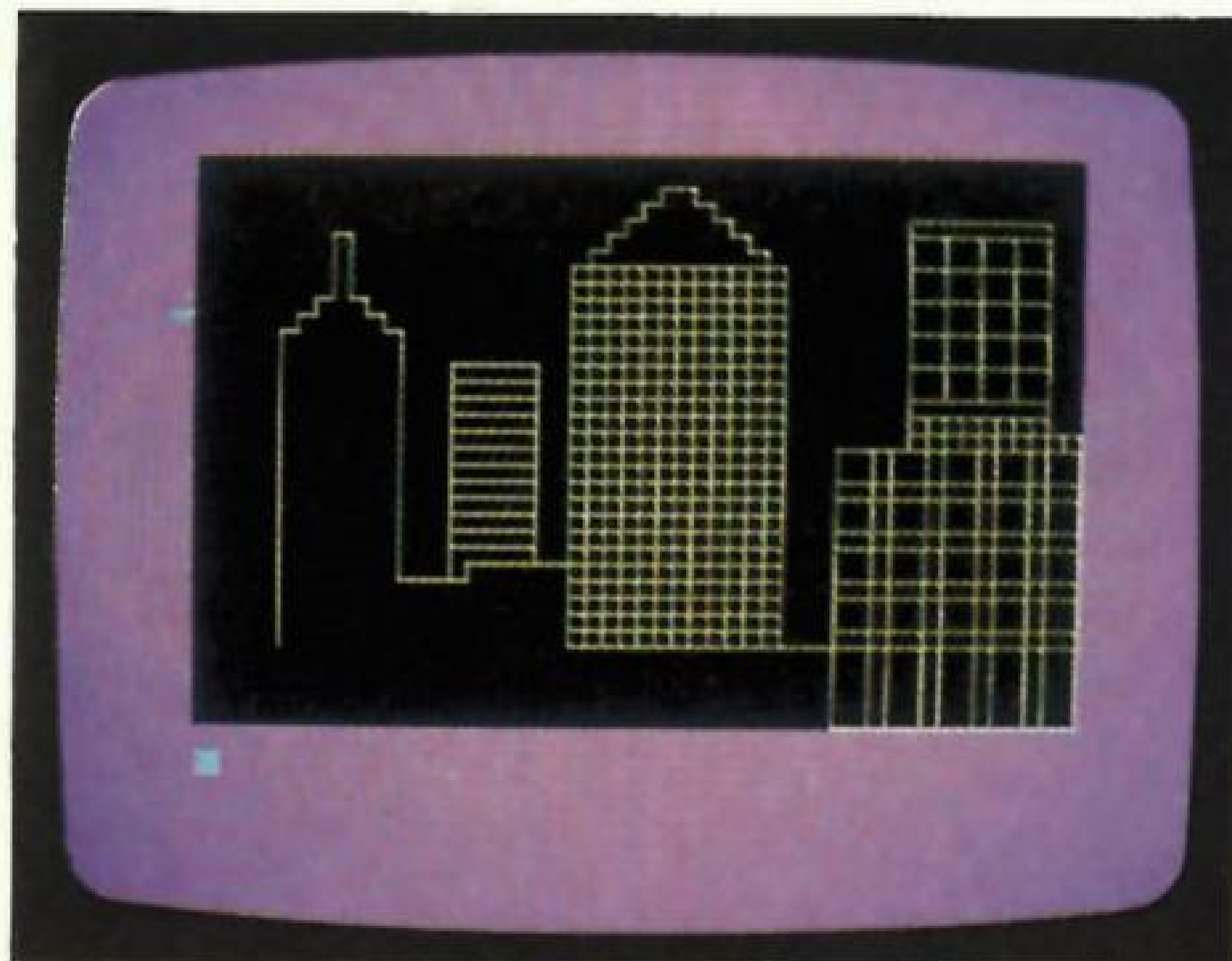
לוח טיוטה על המסך

הספקטרום גמיש מאד. אינך צריך לכתוב תכנית כל אימת שתרצה ליצור תמונה או צורה. במקום זאת, אתה יכול להשתמש בתכנית המאפשרת לך לבנות תמונה ישירות על המסך. להלן מופיעה תכנית פשוטה מאד המאפשרת זאת.

התכנית פותחת במילת המפתח INPUT לקבלת מספר דיו. אחר כך, תוך שימוש נוסף ב־INPUT (הפעם גם סימן $ לסימון מחרוזת), היא גורמת למחשב לצייר קוים קצרים כל אימת שתלחץ על 4 מקשים מוגדרים – u, d, l ו־r.

טיוטה ודוגמה

```
 10 INPUT "INK ";i
 20 BORDER 3: PAPER 0: INK i: C
LS
 30 PLOT 25,25
 40 LET x=5
 50 INPUT k$
 60 IF k$="u" THEN DRAW 0,x
 70 IF k$="d" THEN DRAW 0,-x
 80 IF k$="r" THEN DRAW x,0
 90 IF k$="l" THEN DRAW -x,0
100 GO TO 50
```



קבלת החלטות עם IF ו־THEN

שורות 60 עד 90 בתכנית הטיוטה מכילות הצהרות IF THEN. אלה מאפשרות לספקטרום לקבל החלטות. במקרה זה, המחשב בודק אם המקש עליו לחצת הוא u, d, l או r. אם אחד מאלה נלחץ, אזי המחשב מורה על ציור קו. הוא לא יצייר קו אם תוקש אות גדולה.

IF מלווה תמיד בביטוי שהספקטרום בודק האם הוא אמת או שהוא קיים – כמו מקשים מסויימים שנלחצו. אם הביטוי אמת או קיים, אזי תתבצע הפעולה הבאה אחרי THEN. אם לא, אזי התכנית עוברת לשורה הבאה.

כל מה שבא אחרי THEN בשורה זו כפוף להחלטה. בשורה שלהלן:

```
110 IF b=5 THEN PRINT"*":GOTO 200
```

המחשב יפנה לשורה 200 רק אם b הוא 5.

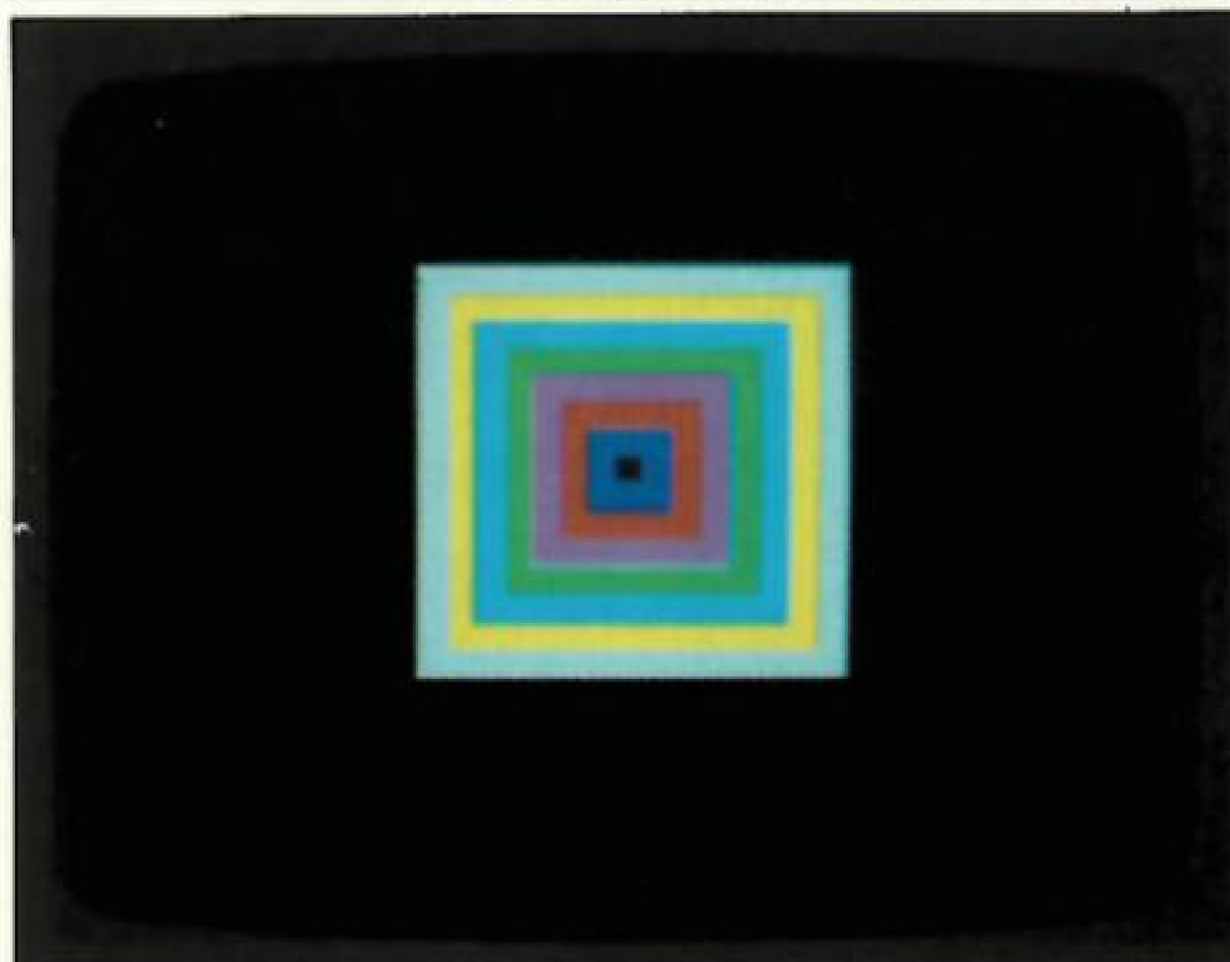
תכנן בעצמך צורות ותמונות שונות

אתה יכול ליצור בעצמך צורות ותמונות שונות בספקטרום +ZX, תוך שימוש בגרפיקה בהפרדה נמוכה, גרפיקה בהפרדה גבוהה, או שתיהן יחד. הדרך הטובה ביותר להתמודד עם גרפיקה היא להתוות את הצורה המבוקשת על העתק רשת הנקודות בעמ' 8Ø. אחר כך בנה את התכנית שתיצור קוים וצורות במקומות הנכונים על המסך.

כדי לצייר צורות ותמונות, ניתן להשתמש לרוב בלולאות FOR NEXT החוזרות על קטע תכנית מספר פעמים מוגדר מראש. בכל פעם משתנים צבעי ומיקומי התוים או השורות, לרוב באופן קבוע. להלן מופיעה תכנית המשתמשת בטכניקה זאת.

ריבועים

```
10 BORDER 0: PAPER 0: CLS
20 FOR x=7 TO .0 STEP -1
30 INK x
40 FOR l=11-x TO 11+x
50 FOR c=16-x TO 16+x
60 PRINT AT l,c;"■"
70 NEXT c
80 NEXT l
90 NEXT x
```



תכנית זו מכילה 3 לולאות FOR NEXT. לולאת ה־x משנה את צבע וגודל הריבועים הגדולים, בעוד שלולאות l ו־c משנות את מיקום השורה והעמודה של הריבוע הקטן בכל פעם שהוא מודפס. נסה לשנות את הריבוע בשורה 6Ø לכוכב או תו אחר כלשהו במקלדת.

אפקטים אקראיים ושגרות

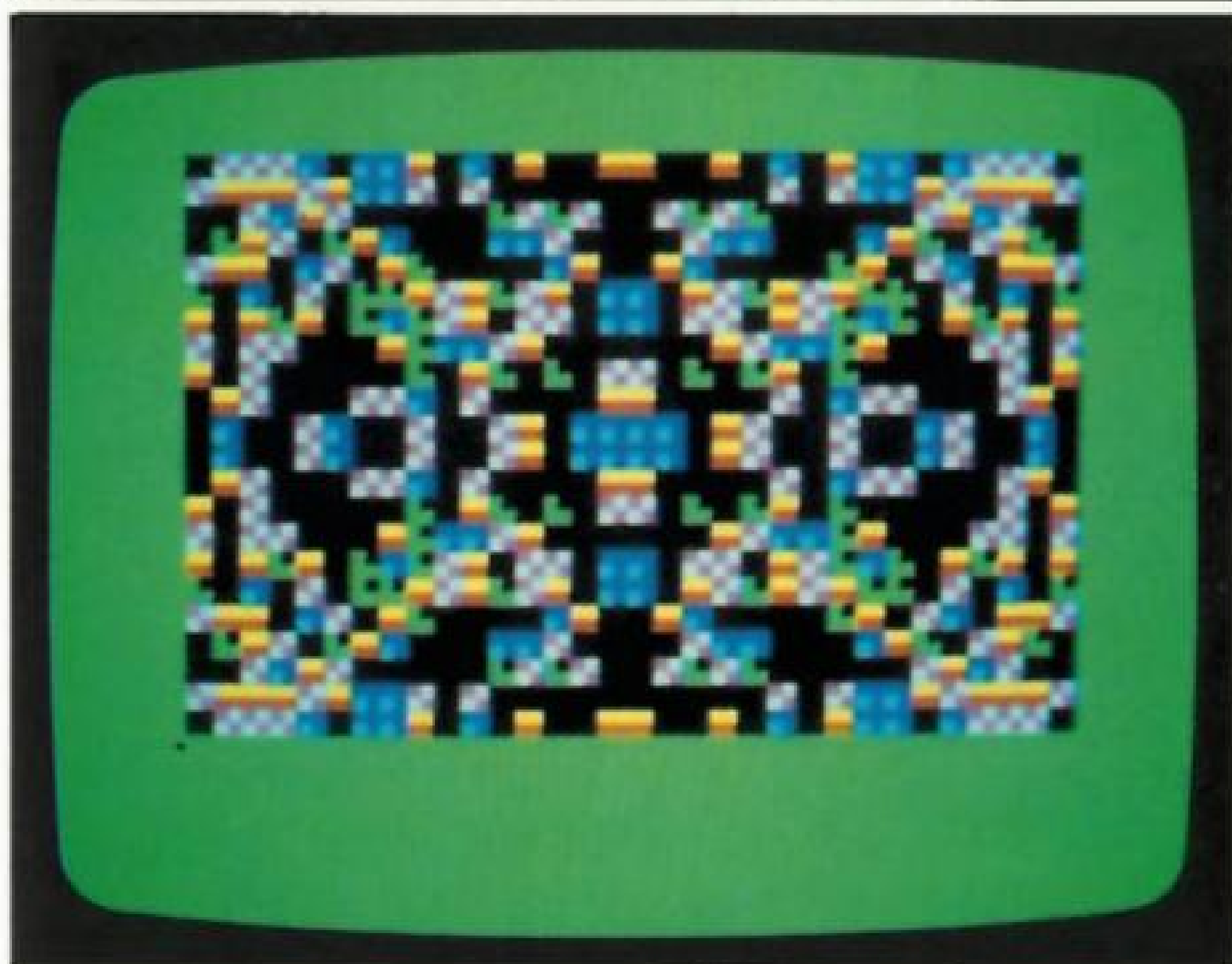
השימוש בלולאות אינו נותן בהכרח צורות זהות בכל פעם שמורצת תכנית גרפית. ע"י הכנסת מילת המפתח RND (קיצור ל־Random — אקראי) בלולאות, ניתן ליצור צבעים, מיקומים ותכונות תצוגה אחרות המשתנות בכל פעם. התבונן בתכנית המוזאיקה בעמ' 1Ø. התכנית פועלת משום שצבע הדיו הוא RND * 7, שמשמעותו כל מספר עם נקודה עשרונית בתחום שבין Ø עד 7. INK משנה זאת למספר השלם הקרוב ביותר. לכן בכל פעם שמוצג ריבוע, צבעו יכול להיות כל צבע שבין INKØ ועד INK7.

התכנית הבאה מציירת צורות סימטריות של תוים גרפיים על המסך. נעשה כאן שימוש ב־RND לשינוי התוים ומיקומם. המשתנים i ו־p נותנים את צבעי הדיו והנייר, ו־a מציין כמה צורות יוצגו (במקרה זה ארבע). המשתנה n נותן את מספר התוים בכל צורה, בעוד ש־x הוא מספר אקראי בתחום 129 עד 142.

הצהרת GOSUB 1ØØØ בשורה 5Ø שולחת את המחשב אל שגרה.

צורות סימטריות

```
10 BORDER 4: PAPER 0: CLS
20 LET i=4: LET p=0
30 FOR a=1 TO 4
40 LET x=RND*13+129
50 FOR n=1 TO 40: GO SUB 1000:
 NEXT n
60 LET i=i+1: LET p=p+1
70 PAUSE 100
80 NEXT a
90 STOP
1000 LET l=INT (RND*11)
1010 LET c=INT (RND*16)
1020 INK i: PAPER p
1030 PRINT AT l,c;CHR$ x
1040 PRINT AT l,31-c;CHR$ x
1050 PRINT AT 21-l,c;CHR$ x
1060 PRINT AT 21-l,31-c;CHR$ x
1070 BEEP 0.01,l*c/3
1080 RETURN
```



שגרה היא קבוצת שורות המתנהגת כתכנית־בתוך תכנית. בתכנית זאת השגרה נמצאת בשורה 1ØØØ. היא מציגה תו גרפי בארבעת רבעי המסך כך שכל אחד מרוחק במידה שווה מהמרכז (נקודה 11,16). מרחק זה נתון בשורות 1ØØØ ו־1Ø1Ø. l נותן את המרחב בשורות ו־c את המרחק בעמודות. INT משנה את המספר האקראי למספר שלם כך ש־l הוא מספר שלם כלשהו בין Ø ל־1Ø ו־c הוא מספר שלם כלשהו בתחום Ø עד 15. שורות 1Ø3Ø עד 1Ø6Ø מציגות את התו הגרפי בעל קוד x (ראה טבלת מערך תוים בעמ' 51). BEEP מייצרת צליל שגבהו השור למיקום הנקודה, ואז RETURN בשורה 1Ø8Ø שולח את התכנית חזרה להצהרה הבאה אחרי GOSUB בשורה 5Ø.

שורה 6Ø משנה את צבעי הדיו והנייר, ואז PAUSE 1ØØ בשורה 7Ø משהה את התכנית למשך 2 שניות לפני השלמת הלולאה וחזרה לנקודת ההתחלה. בשורה 9Ø נדרש STOP לעצירת התכנית לבל תמשיך אל תוך השגרה לאחר הלולאה הרביעית.

אתה יכול לשנות תכנית זו ע״י שינוי הערך 4 בשורה 3Ø, ו־4Ø בשורה 5Ø למספרים אחרים. אם תגדיל יותר את טווח x בשורה 4Ø, תקבל תוים אחרים שיופיעו על המסך. אל תגדיל את טווח ערכי i רק ליותר מאשר 7.

השימוש בלולאות FOR NEXT בגרפיקה

ניתן להשתמש ביעילות בלולאות FOR NEXT בגרפיקה בהפרדה גבוהה ליצירת תמונות הבנויות מקוים וצורות קבועים. הקש פנימה והרץ את התכנית הבאה. על ידי שימוש ב־PLOT ו־DRAW בלבד, שתי לולאות FOR NEXT מציירות בתחילה קוים על הקרקע ואח״כ 5 משולשים או פירמידות מלאות.

פירמידות

```
10 BORDER 0: PAPER 1: INK 6
20 CLS
30 FOR y=0 TO 20 STEP 2
40 PLOT 0,y
50 DRAW 255,0
60 NEXT y
70 FOR n=100 TO 220 STEP 30
80 FOR x=-10-n/10 TO 10+n/10
90 PLOT n,35+n/10
100 DRAW x,-n/4
110 NEXT x: NEXT n
```



כעת הוסף את השורות בתכנית הבאה. כאשר תריץ אותה שנית, תמצא שקרן לייזר נורית ללא הרף לשמי הלילה, ויוצרת צרורות כוכבים. הקרן מצויירת מפינת המסך לנקודה x,y, כאשר x ו־y הם מספרים אקראיים. מספרים אלה מומרים למספרי מיקום כוכבים בהפרדה נמוכה.

```
120>LET x=RND*255
130 LET y=RND*104+71
140 LET l=INT (175-y)/8
150 LET c=INT (x/8)
160 PLOT 0,0: DRAW OVER 1;x,y
170 BEEP 0.01,x/4
180 PLOT 0,0: DRAW OVER 1;x,y
190 PRINT AT l,c;"*"
200 GO TO 120
```



OVER 1 בשורות 16Ø ו־18Ø מאפשר לשורה הראשונה לצייר את קרן הלייזר ולשורה השנייה למחוק אותה ללא שינוי יתר התמונה. שמור תכנית זו מאחר ותזדקק לה במועד מאוחר יותר.

INVERSE ,BRIGHT ,FLASH

שלש מילות מפתח אלה מאפשרות גמישות רבה בהפעלת צבעי הספקטרום. כל מילת מפתח מלווה ב־Ø או 1, ועליך להציבם בהצהרות PRINT בתנאי שיש נקודה־פסיק אחרי ה־Ø או ה־1. FLASH 1 גורמת למקומות תווים להבהב בין צבעי הדיו והנייר, בעוד ש־BRIGHT 1 יוצרת צבעים בהירים יותר. INVERSE 1 משנה את צבע הדיו לצבע הנייר ולהיפך. הצבת Ø אחרי מילות מפתח אלה מחזירה את התצוגה למצב הקודם.

נסה להכניס שינויים אלה בתכנות בפרק זה כדי לראות את פעולת מילות המפתח. בתכנית הריבועים, שנה את הריבוע בשורה 6Ø לכוכב ואז הוסף

```
15 INVERSE 1
```

כעת הכוכבים יופיעו בשחור (צבע הנייר) כנגד פסים צבעוניים (צבעי הדיו המתחלפים). הקש INVERSE Ø לפני שתמשיך.

בתכנית הצורות הסימטריות, הוסף את השורות הבאות כדי לראות את פעולת BRIGHT ו־FLASH.

```
15 BRIGHT 1
16 FLASH 1
```

שים לב כיצד FLASH גורם לצורה להיראות כאילו היא נעה, הקש FLASH Ø : CLS להפסקת הבהוב התצוגה.

כל השינויים האלה משפיעים על התצוגה המלאה שיוצרת כל תכנית. השימוש ב־FLASH, BRIGHT או INVERSE בתוך הצהרת PRINT מגביל את פעולת 3 מילות המפתח להדפסה הנוצרת בשורה זו בלבד.

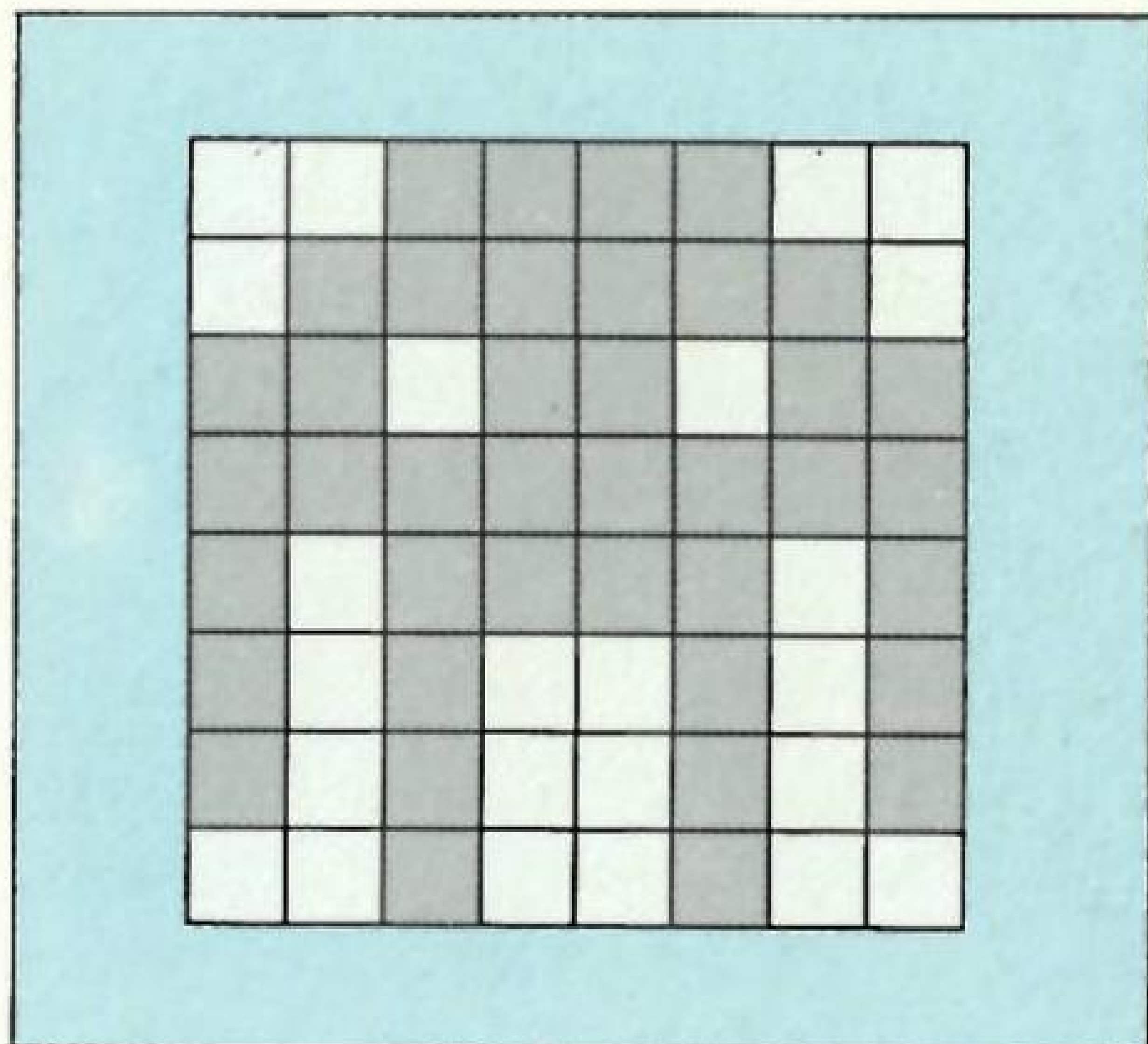
כיצד ליצור תוים משלך במחשב

הספקטרום +ZX אינו מוגבל רק לתוים גרפיים שניתן להקיש מהמקלדת. בחלק מיוחד של הזכרון, ניתן לשמור תוים אחרים שתכננת בעצמך. אלה נקראים תוים גרפיים מוגדרי־משתמש וכל תכנית יכולה להכיל עד 21 תוים כאלה.

כל תו בנוי מעד 64 נקודות קטנות או פיקסלים בצבע דיו. הפיקסלים מסודרים ב־8 שורות בנות 8 פיקסלים כל אחת, וכל תו תופס מקום של תו אחד ברשת הנקודות של הפרדה נמוכה – בדיוק כמו התוים הגרפיים התקניים במקלדת.

עיצוב תו גרפי

ראשית צייר רשת 8×8 כמתואר להלן. מלא בצבע חלק מהריבועים ליצירת תו. רבועים אלה מייצגים את הפיקסלים בצבע דיו. כל רבוע מלא נחשב כערך 1 וכל רבוע ריק נחשב כ־0. התכנית הבאה מתארת בניית עכביש.



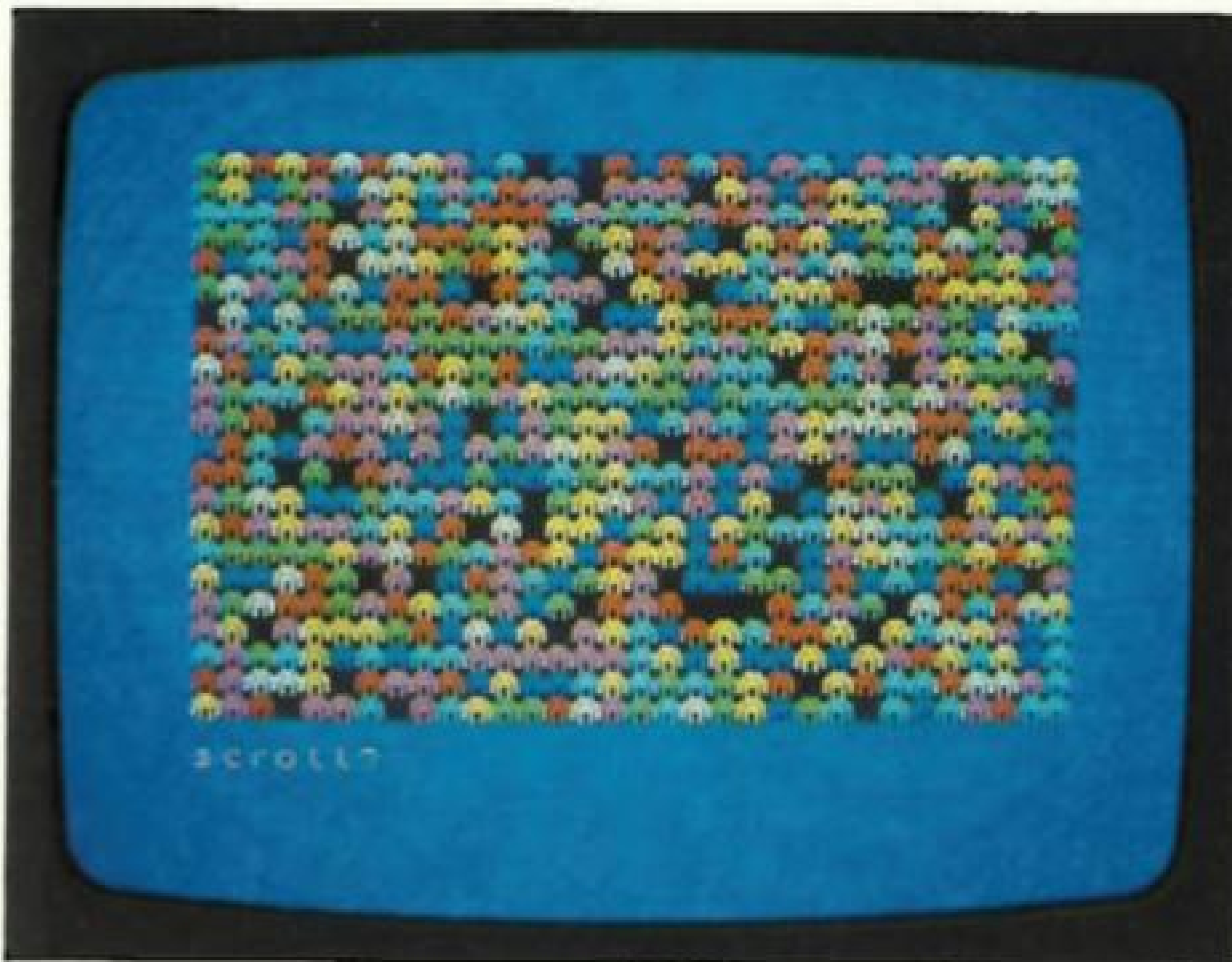
כל אחד מהתוים המוגדרים ע״י המשתמש מזוהה ע״י אות מ־a עד u (או A עד U – אין הבדל). לתיכנות האות, הקש 8 הצהרות POKE USR כשכל אחת מהן מסתיימת ב־BIN בלווי מספר בינארי המורכב משמונת ה־0 או 1 בכל שורה של רשת הנקודות. הבה ונקרא את תו העכביש S.

```
10 POKE USR "s",BIN 00111100
11 POKE USR "s"+1,BIN 01111110
12 POKE USR "s"+2,BIN 11011011
13 POKE USR "s"+3,BIN 11111111
14 POKE USR "s"+4,BIN 10111101
15 POKE USR "s"+5,BIN 10100101
16 POKE USR "s"+6,BIN 10100101
17 POKE USR "s"+7,BIN 00100100
```

כעת הרץ את התכנית ולחץ GRAPH ר־S, מופיע העכביש! עתה הוסף את השורות הבאות, לקבלת העכביש בשורה 30 בדרך זו, והרץ את התכנית. עכבישים יופיעו על פני כל המסך.

עכבישים

```
20 BORDER 1: PAPER 0: CLS
30 PRINT INK RND*7;"♠";
40 GO TO 30
```



כאשר אתה מעצב תוים משלך, זכור שלא תוכל לראותם על המסך אלא אם הרצת תכנית אשר מגדירה אותם. עד אז, הם יופיעו ברישומים כאותיות.

כיצד ל'ערבב' צבעים עם ריבועים מנומרים

אתה יכול לדמות בקלות ערבוב צבעים בספקטרום. כדי לעשות זאת תצטרך ליצור תו שכאשר יודפס יתן רבוע שהוא 50% בצבע דיו ר־50% בצבע הנייר.

```
  10 FOR x=0 TO 6 STEP 2
  20 POKE USR "a"+x,BIN 10101010
  30 POKE USR "a"+x+1,BIN 010101
01
  40 NEXT x
```

עליך רק להגדיר שתי שורות פיקסלים ולהורות למחשב להשתמש בהן לחילופין בתוך תו.

כאשר תריץ את התכנית, תראה רבוע מנומר. אם תנקוט באותה טכניקה בתכנית המכילה מילות מפתח לצבעים, הרבוע המנומר יצור צבע שיהיה תערובת של צבעי הדיו והנייר בתכנית.

פישוט הגרפיקה בעזרת READ ו־DATA

להלן מוצעת דרך קלה יותר ליצירת גרפיקה במחשב בעזרת שימוש במספרים עשרוניים בפקודות READ ו DATA. ראשית שנה את 8 המספרים הבינאריים הבנויים מ־0 ו־1 למספרים עשרוניים. עשה זאת בעזרת הקשת PRINT BIN המלווה במספר. לדוגמה:

```
PRINT BIN 00111100
```

הספקטרום יציג 60, שהוא אקוויולנטי למספר הבינארי 00111100. במקרה של העכביש, 8 המספרים העשרוניים הם 60, 126, 219, 255, 189, 165, 165 ו־36. כעת אתה יכול להשתמש ב־READ ו־DATA. שתי מילות מפתח אלה נותנות דרך נוחה להזנת ערכים רבים כמו מספרים למשתנים בתכנית. READ מלווה במשתנה — אות אחת או יותר אם עוסקים במספרים, או אות בודדת בלוויית $ במקרה של מחרוזות.

מכיוון שכאן תרצה לקרוא מספרים עשרוניים, בחר במשתנה מספרי שיקרא y.

כאשר הספקטרום נתקל במלה READ, הוא פונה להצהרת DATA הראשונה בתכנית. הצהרה זאת כוללת רשימת ערכים המופרדים בפסיקים. המחשב לוקח את הערך הראשון ברשימה ונותן אותו למשתנה הבא אחרי READ. הערך השני ינתן למשתנה הבא, וכן הלאה לפי הסדר.

להלן מופיעה התכנית החדשה ליצירת העכביש.

```
10 FOR x=0 TO 7
20 READ y
30 POKE USR "s"+x,y
40 NEXT x
50 DATA 60,126,219,255,189,165
,165,36
```

התכנית תאחסן למעשה בזכרון 8 מספרים עשרוניים כלשהם ליצירת תו. עליך רק לשנות את s בשורה 30 לאות שתרצה, ואחרי DATA בשורה 50 תקיש את 8 המספרים, כשהם מופרדים בפסיק. לחץ GRAPH ואת האות לקבלת התו לאחר שהרצת את התכנית.

ציור לוח שחמט

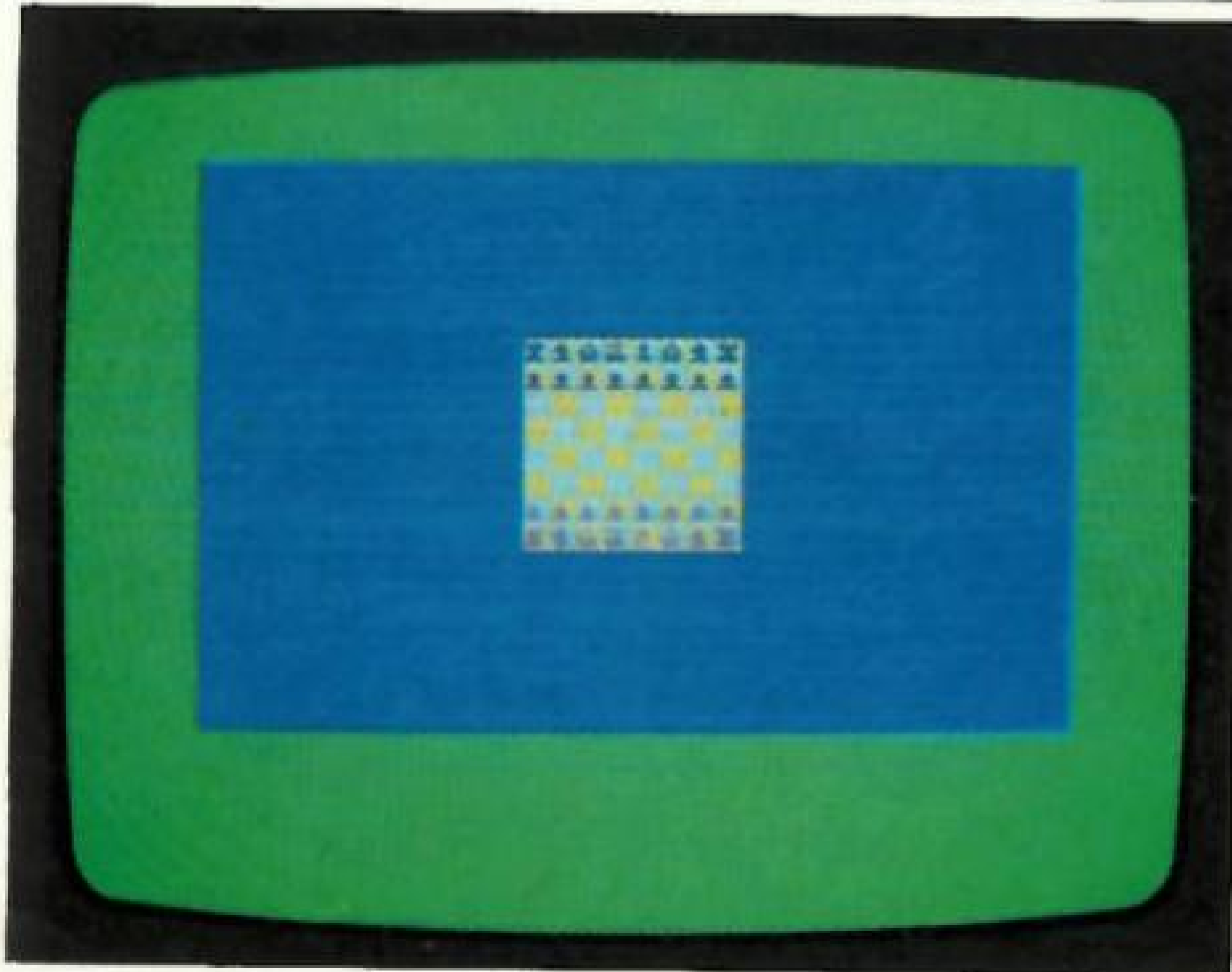
כאן מוצעת תכנית המציגה לוח שחמט על המסך ועורכת עליו את הכלים לקראת משחק. אתה יכול ליצור צבעים שיופיעו ברשימה בעזרת קודי בקרת צבע — ראה את הטבלה שלהלן.

לוח שחמט

```
10 FOR x=1 TO 6
20 READ a$
30 GO SUB 500
40 NEXT x
50 BORDER 4: PAPER 1: CLS
60 FOR l=7 TO 14 STEP 2: FOR c
=12 TO 19 STEP 2
70 PRINT AT l,c;" ";AT l+1,c;
" "
80 NEXT c: NEXT l
90 PRINT AT 7,12;"RNBQKBNR"
100 PRINT AT 8,12;"PPPPPPPP"
110 PRINT AT 13,12;"PPPPPPPP"
120 PRINT AT 14,12;"RNBQKBNR"
130 GO TO 130
500 FOR n=0 TO 7
510 READ y
520 POKE USR a$+n,y
530 NEXT n
540 RETURN
```

הכלים מוגדרים בשגרה המתחילה בשורה 500.

```
550 DATA "p",0,0,16,56,56,16,12
4,0
560 DATA "r",0,84,124,56,56,124
,124,0
570 DATA "n",0,16,56,120,24,56,
124,0
580 DATA "b",0,16,40,68,108,56,
124,0
590 DATA "K",0,16,56,16,56,68,5
6,0
600 DATA "q",0,84,40,16,108,124
,124,0
```



טבלת שימוש בקודי בקרת צבע

במקום להשתמש במילות מפתח כמו INK ו־PAPER, ניתן להציב קודי בקרה בתוך הצהרות PRINT אחרי סימן המרכאות הראשון. התוים המיועדים לתצוגה יחליפו את צבעיהם ברישום עצמו ויופיעו בצבעים אלה גם על המסך.

כדי להגיע לקודים, לחץ תחילה על EXTEND MODE ואח"כ על מקש מספר עם או בלי CAPS SHIFT. טבלה זו מראה כיצד תוכל לבחור כל אחד מהצבעים.

לחץ	1	2	3	4	5	6	7	8	9	0
EXTEND MODE + CAPS SHIFT	כחול דיו	אדום דיו	סגול דיו	ירוק דיו	תכלת דיו	צהוב דיו	לבן דיו	הבהוב לא פועל	הבהוב פועל	שחור דיו
EXTEND MODE	כחול נייר	אדום נייר	סגול נייר	ירוק נייר	תכלת נייר	צהוב נייר	לבן נייר	בהירות לא פועל	בהירות פועל	שחור נייר

הנפשה

גרפיקה במחשב נראית במיטבה כאשר התוים או השורות נעים על המסך, ויצירת הנפשה בספקטרום אינה קשה. כל מה שעליך לעשות הוא להמשיך ולשנות את מיקום הדפסת תו או ציור שורה. הדרך הטובה ביותר לצורך זה היא בעזרת לולאה אחת או יותר של FOR NEXT.

תנועה אנכית ואפקית

הקש פנימה והרץ תכנית זאת. אם לא כבית את הספקטרום או לחצת על כפתור RESET מאז יצירת תו העכביש בעמ' 32, אל תטרח להקיש שורות 10 עד 50. התו הגרפי ישמר עדיין בזכרון תחת "S".

עכביש נופל

```
  5 BORDER 3: PAPER 5: CLS
 10 FOR x=0 TO 7
 20 READ y
 30 POKE USR "s"+x,y
 40 NEXT x
 50 DATA 60,126,219,255,189,165
165,36
 60 FOR x=0 TO 7
 70 READ y
 80 POKE USR "t"+x,y
 90 NEXT x
100 DATA 16,16,16,16,16,16,16,1
6
110 FOR l=0 TO 20
120 PRINT AT l,3; INK 0;"|"
130 PRINT AT l+1,3; INK 2;"■"
140 NEXT l
```

בכל פעם שתריץ את התכנית תוכל לראות את העכביש נופל לאורך המסך כשהוא תלוי על קוריו.

שורות 60 עד 100 בתכנית יוצרות תו גרפי אחר ("t") עבור הקור. ההנפשה מתבצעת בשורות 110 עד 140, היוצרות לולאת FOR NEXT שבה l (מספר השורה) משתנה מ־0 עד 20. בכל פעם שהלולאה חוזרת על עצמה, אורך קור מודפס בנקודה אחת והעכביש מודפס בנקודה הבאה מתחתיו. בפעם הבאה, העכביש מוחלף באורך קור נוסף והעכביש מופיע שוב מתחתיו. בצורה זו, העכביש משתלשל מטה במהירות לאורך הקור עד שהוא מגיע לשורה 21, שהיא תחתית שטח התצוגה.

הספקטרום מסוגל לחשב את מיקום הנקודות הבאות במהירות רבה, כך שהעכביש צונח במהירות. כדי להאיט את תנועתו, הכנס שורה זאת:

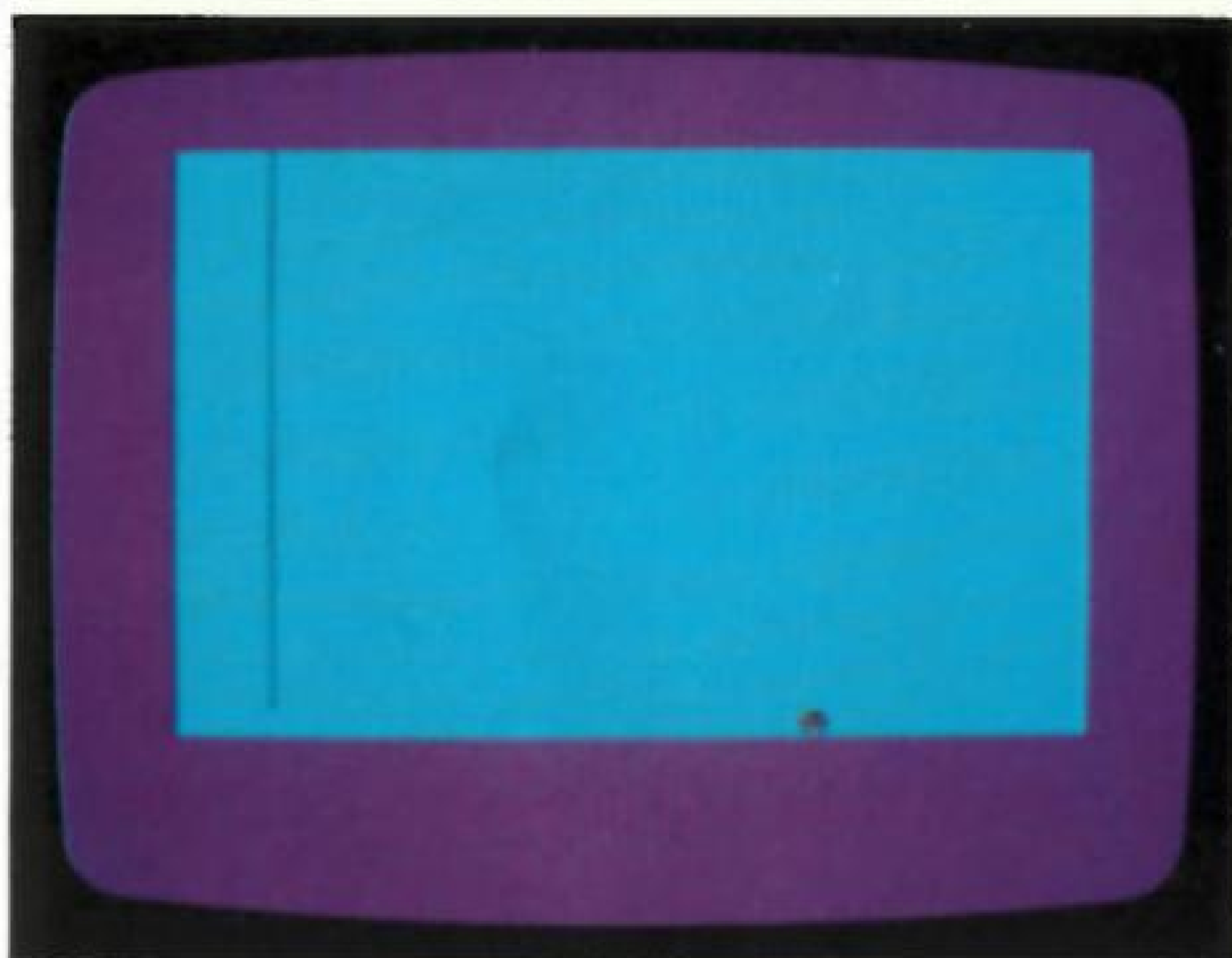
```
135 PAUSE 10
```

פקודה זו גורמת למחשב להמתין חמישית שנייה בכל פעם לפני הדפסת העכביש בנקודה הבאה. נסה לשנות 10 לערכים אחרים וראה כיצד משתנה המהירות.

כעת הוסף שורות אלה לתכנית והרץ אותה שנית. תוכל לראות הנפשה אך הפעם בכוון שונה.

עכביש מתרוצץ

```
150 FOR c=3 TO 30
160 PRINT AT 21,c;" "
170 PRINT AT 21,c+1; INK 2;"■"
180 NEXT c
```



כעת העכביש מתרוצץ הצידה ברגע שהגיע ארצה. השורות הנוספות יוצרות לולאת FOR NEXT נוספת המבקרת את מיקום העמודה c. שים לב לכך שקודם מודפס רווח ואז מוצג העכביש במיקום העמודה הבאה. כך נוצר מצב בו העכביש נעלם מנקודה אחת ומופיע בנקודה הבאה בכוון ימין. תמיד עדיף למחוק תו לפני הדפסתו במיקום הבא, מאשר להדפיס קודם את התו במיקום החדש ורק אז למחקו מהמיקום הישן. כך מונעים או מפחיתים הבהוב בגרפיקה נעה.

אימון קליעה למטרה

במשחקי מחשב רבים, מתרחשת לרוב פעולה כאשר שני גופים נעים מתנגשים או שגוף נחבט בקיר. כיצד יודע המחשב מתי אמורה להתרחש התפוצצות או התנגשות?

זיהוי התנגשויות אינו קשה. אם שני תוים מודפסים בנקודה l,c (עבור שורה ועמודה) ומיקום v,h (עבור מיקום אנכי ואפקי) אזי אם l = v רח c = h, שניהם חייבים להיות באותו מקום. ניתן לכתוב זאת כהצהרה, לדוגמה:

```
160 IF l=v AND c=h THEN PRINT"CRASH!"
```

דרך אחרת לזיהוי התנגשויות היא בעזרת צבע. מחק את תכנית העכביש בהקשת NEW. הקש פנימה את תכנית הפירמידות בעמ' 31 במלואה, או טען את התכנית אם שמרת אותה על סרט. אתה יכול עתה לשפרה ולשלבה עם העכביש (הנמצא עדיין בזכרון אלא אם כבית את המחשב או לחצת RESET) ליצירת תכנית אחרת.

ראשית הוסף את השורות הבאות, היוצרות גרפיקת התפוצצות הנקראת "e".

```
5 FOR x=0 TO 7
6 READ y
7 POKE USR "e"+x,y
8 NEXT
9 DATA 145,82,44,121,158,52,7
4,137
```

כעת מחק שורה 190 והוסף או שנה את השורות הבאות.

עכבישים ופירמידות

```
114 LET h=RND*31
115 FOR v=0 TO 20
117 PRINT AT v,h;" ";AT v+1,h;
INK 4;"S"
200 NEXT v
205 PRINT AT 21,h; FLASH 1; INK
2; PAPER 6;"S"
210 GO TO 114
```



צרורות הכוכבים לא מופיעים עוד. במקומם נופלים מהשמיים עכבישים המכרסמים את הפירמידות והקרקע. מה שביצעת הוא הוספת לולאת FOR NEXT אשר בה v ו־h נותנים את מיקום העכביש. המשתנה h הוא אקראי, כך שהעכבישים מתחילים את צניחתם האנכית במקומות שונים על המסך. כעת הוסף שורות אלה.

עכבישים מתפוצצים

```
190 IF ATTR (v+1,h)=14 THEN GO
TO 500
500 PRINT AT v+1,h; FLASH 1; PA
PER 2;"E"
510 PAUSE 100
520 GO TO 114
```



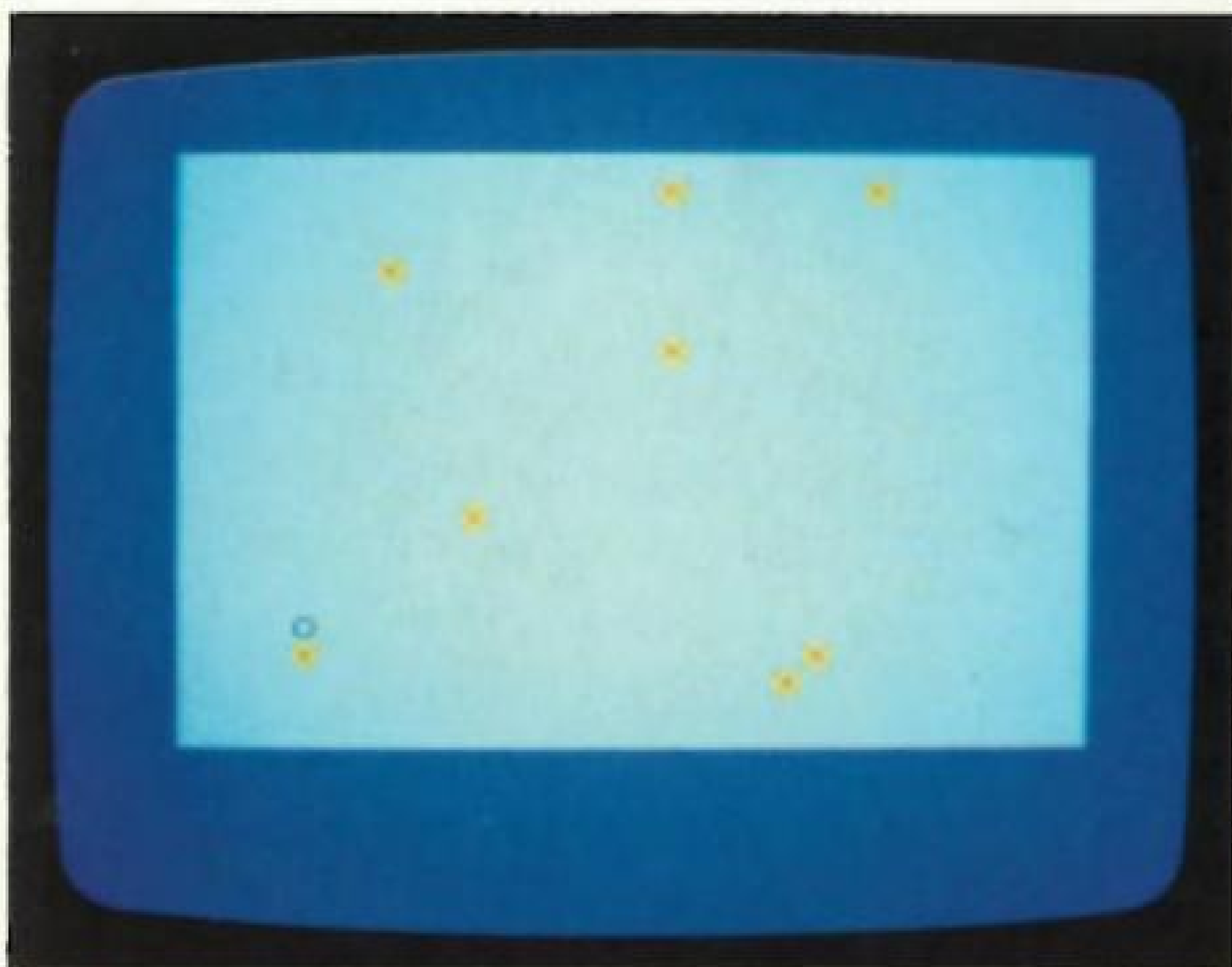
כאשר קרן הלייזר פוגעת בעכבישים, הם הופכים לרגע לצהובים. כאשר הקו הנוצר ע"י DRAW חודר למיקום תו העכביש, צבע הדיו משתנה לצבע השורה, שהוא צהוב. בשורה 190, ATTR מגלה אם העכביש מצהיב ושולח את המחשב לשיגרת ההתפוצצות הנמצאת בשורה 500.

הקפצת כדור

תכניות גרפיות רבות מתארות צורות המנתרות מקצות המסך. תכנית זו מראה לך כיצד הדבר מתבצע. המשתנים v ו־h פועלים באותה צורה כמו תכנית העכבישים המתפוצצים, אך כאן נוסף 1+ או 1 – ל־v או h כדי להניע את הכדור מעלה או מטה, ושמאלה או ימינה. SCREEN $ בודק האם קיים x בנקודה v,h:

כדור מנתר

```
10 BORDER 1
20 FOR z=1 TO 10
30 LET h=INT (RND*25): LET v=I
NT (RND*21)
40 PRINT INK 2; PAPER 6; FLASH
1;AT v,h;"X"
50 NEXT z
60 LET x=1: LET y=1
70 PRINT AT v,h;" "
80 LET v=v+y: LET h=h+x
90 IF h=0 OR h=31 THEN LET x=-
x: BEEP .2,24
100 IF v=0 OR v=21 THEN LET y=-
y: BEEP .2,12
110 IF SCREEN$ (v,h)="X" THEN P
RINT INK 1; PAPER 5;AT v,h;"!*!*
.!": STOP
120 PRINT AT v,h;"O"
130 PAUSE 2
140 GO TO 70
```



שימוש במאפיינים

מילת המפתח ATTR מגלה את ה"מאפיינים" בנקודה מסוימת על המסך. המאפיינים הם צבעי הדיו והנייר והאם הנקודה מהבהבת או בהירה. בתכנית העכבישים המתפוצצים, ATTR מבטיחה שהעכביש מושמד אם הוא מצהיב. זהו, איפוא, צבע הדיו שלו (מספר 6). צבע הנייר הוא כחול (מספר 1) והעכביש אינו מהבהב או בהיר. זה אומר שסה"כ המאפיינים הוא 14. ההסבר על ATTR במדריך העיון למתכנת מתאר כיצד מגיעים למספרים אלה.

כיצד ליצור מוסיקה ואפקטים קוליים

לספקטרום +ZX יש סינתסייזר קולי היכול לעורר את תכניותיך לחיים עם מגוון גדול של קולות מוסיקליים ואפקטים קוליים מיוחדים. הוא פשוט לשימוש, אפילו אם אין לך ידע רחב במוסיקה. הסינתסיזר מפיק אות קולי המועבר לרמקול הפנימי בספקטרום.

תיכנות קולות

כדי להפיק צליל בספקטרום, עליך להשתמש במילת מפתח אחת בלבד – BEEP. היא מלווה בשני מספרים או משתנים המייצגים מספרים. הראשון מורה למחשב מהו משך (בשניות) הצליל, והשני מודיע מהו גובה הצליל. גובה הצליל נמדד ביחידות חצאי־טונים. ערכי הגובה הם 0 עבור דו אמצעי, 1 עבור דו דיאז, 1 – עבור סי נמוך (C_b) וכן הלאה.

כדי לשמוע את טווח הצלילים המלא שהספקטרום מפיק, הרץ תכנית זאת.

הספקטרום עובר דרך כל טווח הטונים מהצליל הגבוה ביותר (69) ועד הנמוך (60 –), תמצא שהצלילים הגבוהים ביותר בקושי נשמעים ואילו הנמוכים נשמעים כנקישות. הסיבה לכך היא שצלילים אלה נמצאים על סף יכולת השמיעה של אוזן האדם.

```
10 FOR p=69 TO -60 STEP -1
20 BEEP 0.2,p
30 PRINT AT 0,0;"   ";AT 0,0;p
40 NEXT p
```

הטבלה שלהלן מראה את ערכי גובה הצלילים כך שבעזרתה תוכל להעביר קטע מדף תווים מוסיקליים לתכנית בספקטרום.

אפקטים קוליים

ניתן להפיק מהספקטרום כל מיני אפקטים קוליים, לרוב ע"י הצבת BEEP בתוך לולאה המשנה במהירות את ערך גובה הצליל. נסה תכניות אלה ושנה אותן כדי לפתח צלילים משלך.

שים לב לכך שערכי משך הצליל קצרים מאד, ומגיעים עד מאית שנייה. לחץ על BREAK לעצירת תכניות הלולאות.

בעבוע

```
 10 LET p=INT (RND*40) -30
 20 BEEP 0.05,p: BEEP 0.05,p+7:
BEEP 0.05,p+4
 30 GO TO 10
```

תכנית זו מנגנת קבוצה של 3 תווים שוב ושוב בגובה צליל אקראי. טווח גובה הצליל רחב, אך תוכל לשנות את הערכים בשורה 10 כדי לשנותו.

מכונה

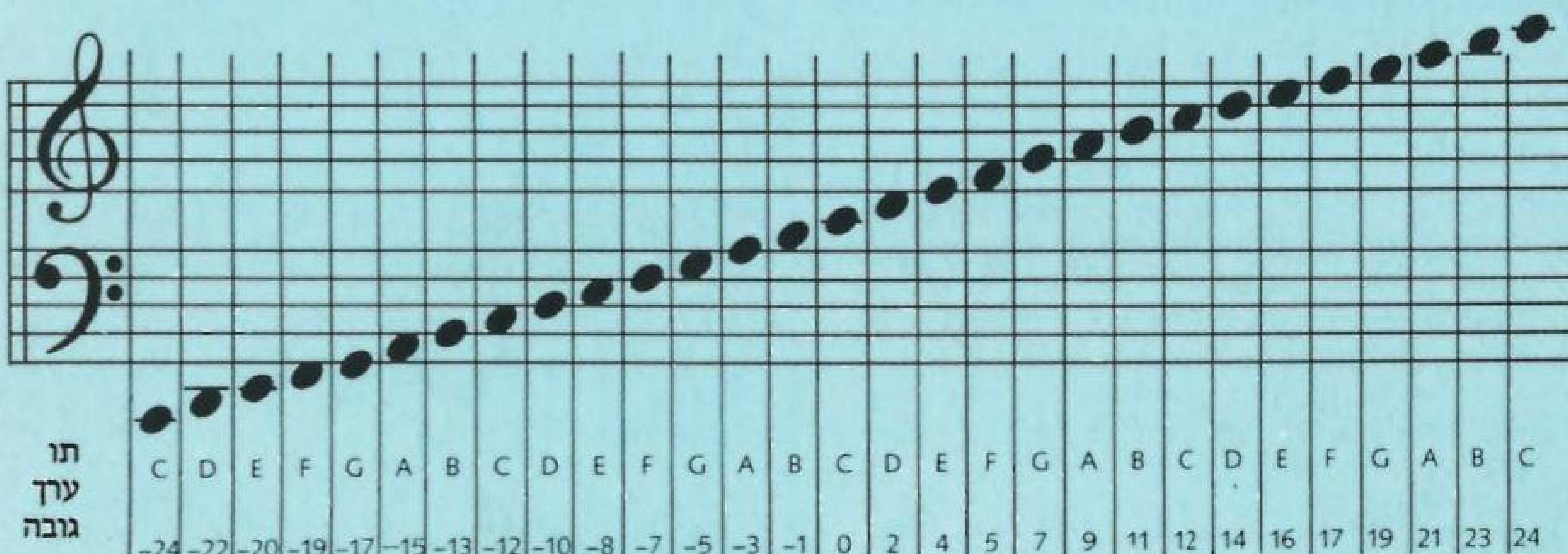
```
10 FOR x=12 TO 36
20 BEEP .01,x
30 BEEP .01,24-x
40 NEXT x
50 GO TO 10
```

תכנית זו מפיקה שני צלילים, האחד עולה בגובה בעוד השני יורד. זה נובע מהצהרות BEEP המשמיעות שני צלילים שוב ושוב בהפרש של מאית שניה ביניהם בגבהים שונים.

טבלת ערכי גובה צליל ליצירת מוסיקה

להלן מופיעים ערכי גובה הצליל בספקטרום מהנמוכים ועד הגבוהים.

הוסף 1 לערך גובה הצליל לקבלת צליל חד; הפחת 1 לקבלת צליל שטות.



התרוממות

```
10 FOR p=1 TO 48 STEP 0.2
20 BEEP .01,p: BEEP .01,p-6
30 NEXT p
```

תכנית זו דומה לתכנית המכונה, אך כעת שני צלילים עולים יחד בהפרש של 6 חצאי־טונים, בנוסף, ערכי גובה הצליל משתנים בצעד של 0.2 — חמישית של חצי־טון — בכל פעם. זה גורם לצליל לעלות בגובה באיטיות. נסה שינויי גובה צליל קטנים אחרים תוך שינוי ערך STEP.

ממיר מקלדת

```
10 LET p=CODE INKEY$
15 IF p=0 THEN GO TO 10
20 BEEP .04,(p-30)/2
30 GO TO 10
```

תכנית זו ממתינה שתקיש על מקש כלשהו. כאשר תעשה זאת, כל אחד יפיק צליל אחר. שים לב שלחיצת CAPS SHIFT תוך לחיצה מטה על מקש אחר גורמת לצליל לרדת. תכנית זו פועלת משום ש־CODE INKEY$ נותן ערך אחר ל־p כל אימת שמקש חדש נלחץ. השורה השנייה עוצרת את המחשב מהשמעת צלילים אם לא נלחץ כל מקש. ניתן לראות את הערכים ש־CODE מחזיר בטבלת מערך התוים בעמ׳ 51.

צליל ומראה

האפקטים הקוליים שהספקטרום מפיק נשמעים היטב על רקע תנועה במסך. כדי להדגים כיצד תוכל להוסיף קול לתכניות, חזור לתכנית העכביש המתרוצץ בעמ׳ 34.

זכור שהכנסת הצהרת PAUSE בשורה 135 כדי להאיט את התנועה. במקום להשהות את התכנית בדרך זו, תוכל לתכנת השהייה המפיקה צליל. שנה שורה 135 לשורה הבאה:

```
135 GOSUB 500
```

כעת הוסף את השורות הבאות לתכנית.

```
200 STOP
500 FOR p=40-l TO 38-l STEP -1
510 BEEP 0.02,p
520 NEXT p
530 RETURN
```

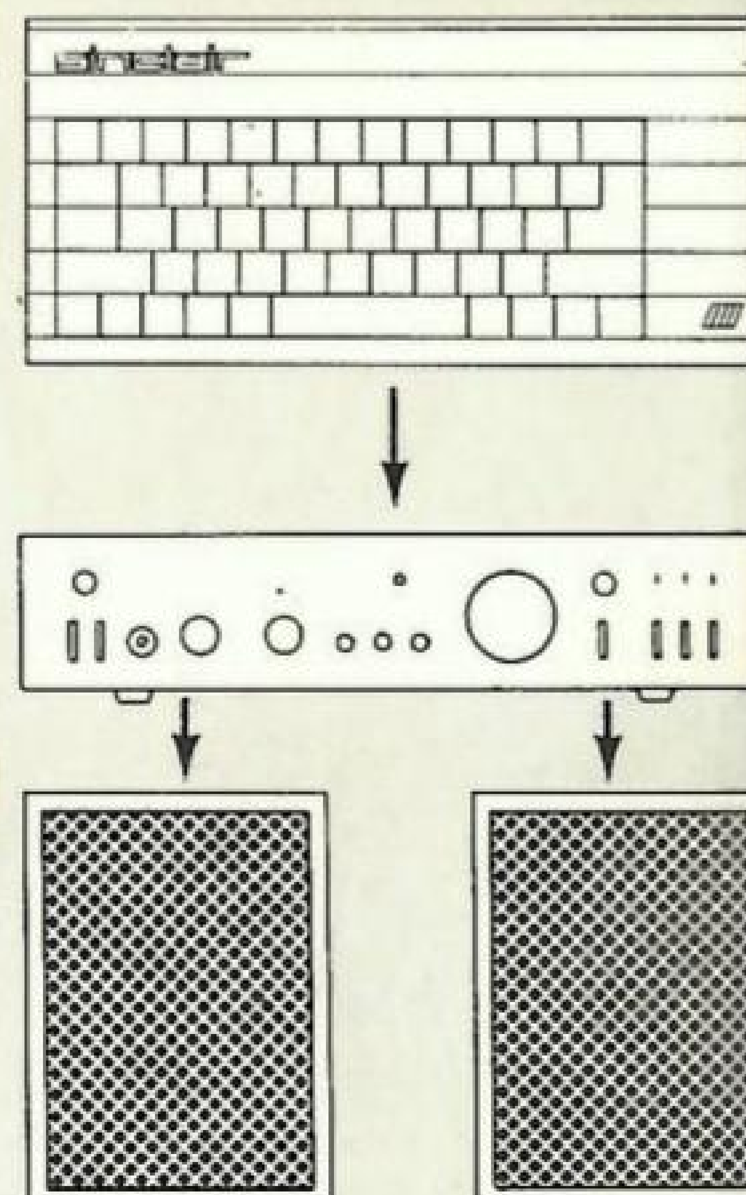
הרץ את התכנית והעכביש יוציא צליל מסולסל בעת צניחתו. השיגרה מנגנת 3 צלילים במהירות רבה הנעשית איטית ככל שהעכביש צונח לנקודה נמוכה יותר במסך. נסה להוסיף תוים נוספים תוך שינוי שורה 500 והאצה או האטת הצלילים ע״י שינוי הערך 0.02 בשורה 510.

כיצד להגביר את הקול

כדי להגביר את עצמת הקול בספקטרום, תוכל לחבר את שקע EAR לאזניות או למגבר ורמקול. למגבר יהיה בקר עצמת קול בעזרתו תוכל לוסת את עצמת הצלילים.

הדרך הפשוטה ביותר לעשות זאת היא להשתמש בכבל הקלטת של הספקטרום לצורך חיבור שקע EAR לשקע MIC של הרשמקול. הוצא החוצה את הקלטת במידת הצורך, הפעל את הרשמקול ואז לחץ PLAY, או FAST ,REWIND (REVERSE) FORWARD (CUE). וסת את עצמת הקול ברשמקול ותוכל לשמוע את צלילי המחשב בוקעים מהרמקול ברשמקול. לחילופין, תוכל לחבר אזניות לרשמקול, אם תרצה.

אתה יכול גם לחבר את הספקטרום למערכת שמע גדולה אם תרצה צליל מלא באמת. כאן תצטרך כבל מיוחד עם תקע פין 3.5 מ״מ המתאים לספקטרום ותקע שיכנס לשקע הקלט במגבר או מערכת השמע. הספקטרום מפיק אות קו הדומה לזה המופק מנגני קלטות (Cassette deck) ורשמי סרט, כך ששקע REPLAY או LINE IN במגבר אמור לפעול. אם מתעוררות בעיות כלשהן, התיעץ בחנות לציוד שמע.





כיצד לשמור את תכניותיך

לא יעבור זמן רב, ותרצה לשמור את תכניותיך על סרט קלטת. כדי לעשות זאת, עליך לחבר רשמקול קסטות לספקטרום ולשמור את התכנית הנמצאת במחשב. הספקטרום שולח את התכנית לרשמקול הקלטות בצורה בה ניתן להקליטה על סרט. ואז, כאשר תרצה להשתמש בתכנית, תוכל לטעון אותה מהרשמקול למחשב בעזרת תהליך הטעינה המתואר בעמ' 15-14. בפרק שלהלן תוכל לראות כיצד לשמור תכניות ואיך לוודא שהתכנית אכן נשמרה כראוי.

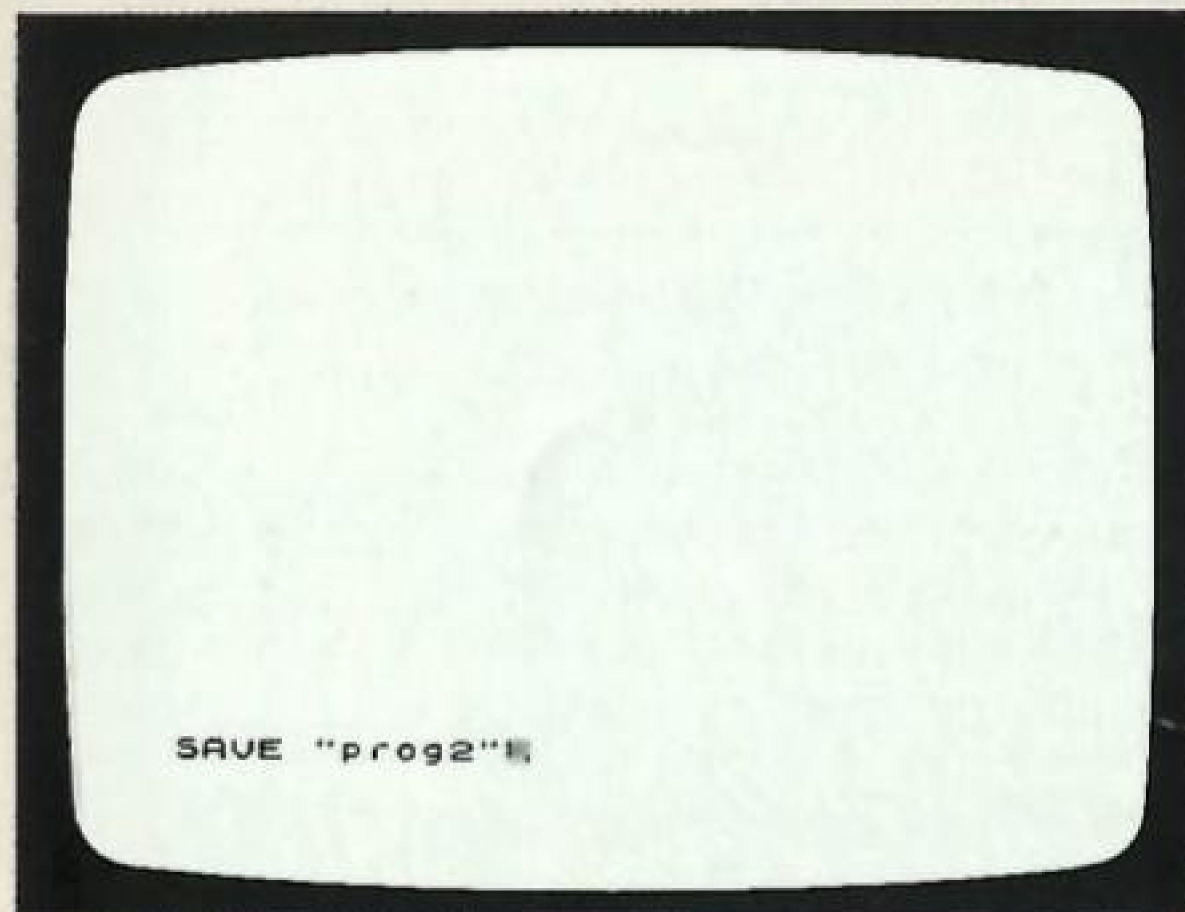
הקלטת תכניות על סרט

1 ראשית חבר את הספקטרום לרשמקול קלטות מתאים תוך שימוש בכבל הקלטת כמתואר בעמ' 14, אך וודא שרק שקע MIC בספקטרום אכן מחובר לרשמקול.

2 אם יש בקרי עצמת הקלטה ועצמת קול ברשמקול, ווסת אותם לשני שליש מהעצמה המירבית. אם לא, אל תדאג מכיוון שעצמת ההקלטה תווסת באופן אוטומטי.

3 הקש פנימה SAVE מלווה בשם התכנית בין מרכאות, לדוגמה:

SAVE"prog 2"

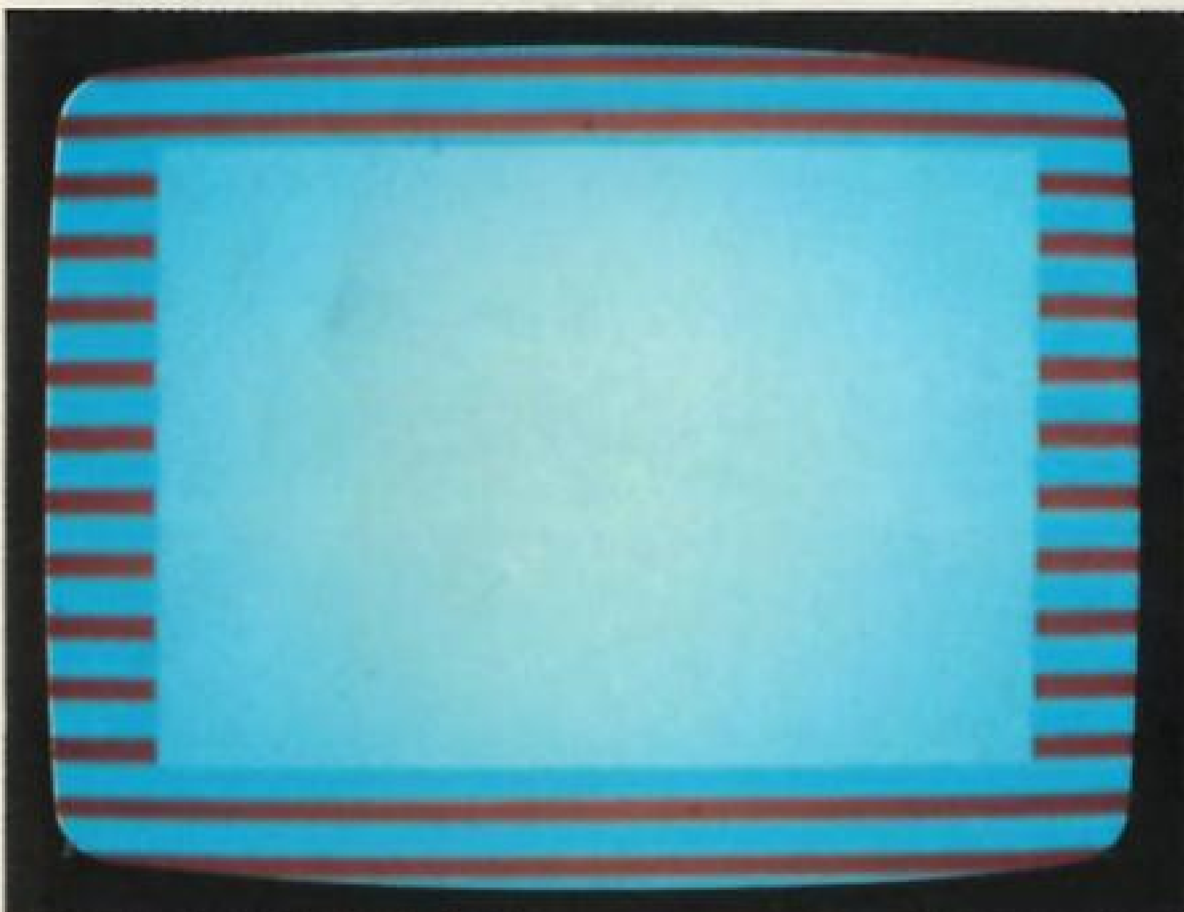


ניתן להשתמש בכל צירוף שהוא של עד 10 אותיות ומספרים. כעת לחץ ENTER. הקו של SAVE יעלם ואז תוכל לראות את הוראת הפעלת הקלטת מתוך הספקטרום.



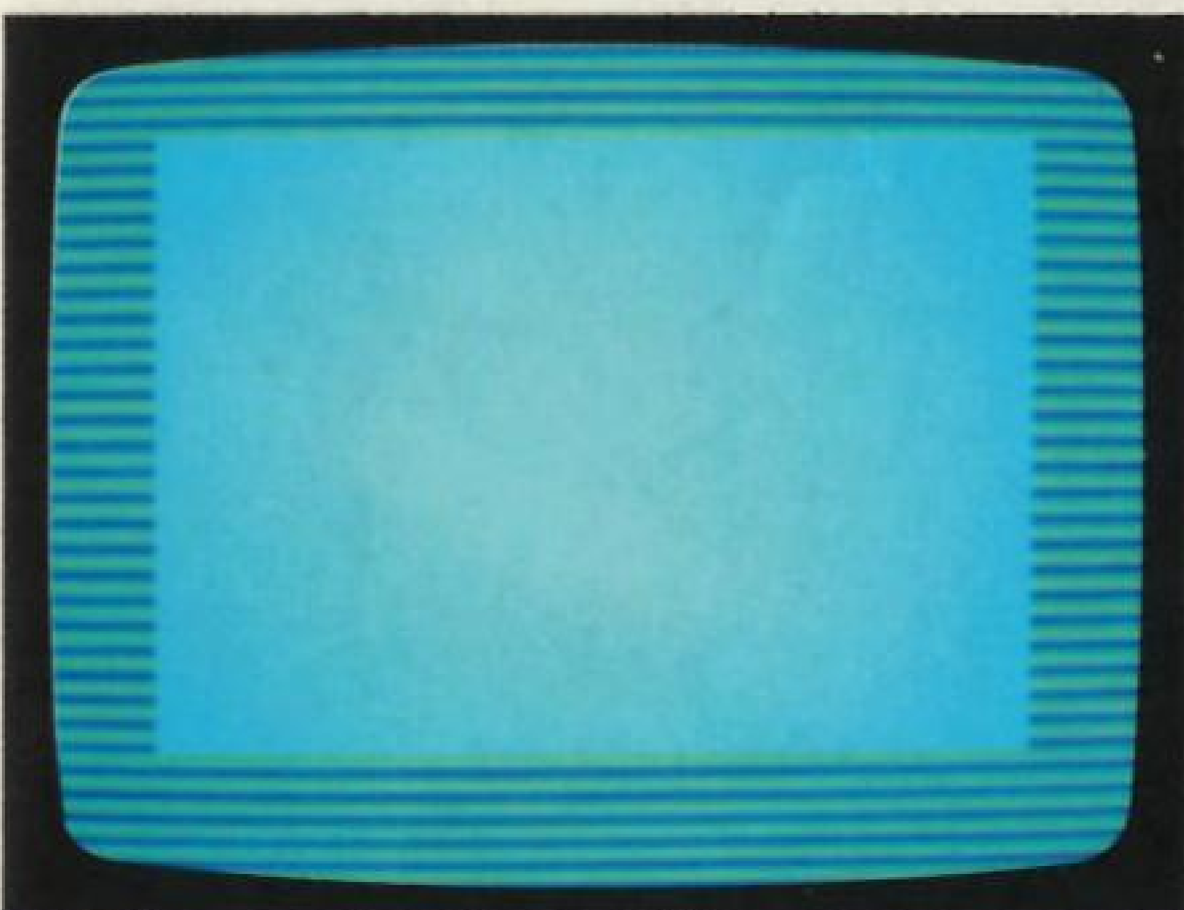
4 הצב את הרשמקול במצב הקלטה, בעזרת לחיצה על RECORD ו־PLAY יחד. אחר כך לחץ על מקש כלשהו בספקטרום.

5 כעת חכה עד שהספקטרום יסיים את העברת התכנית. ראשית תראה פסים כחולים ואדומים נעים באיטיות על המסך.



אחר כך יופיע פרץ קצר של פסים כחולים וצהובים. זה קורה בזמן שהספקטרום שולח את שם התכנית לסרט.

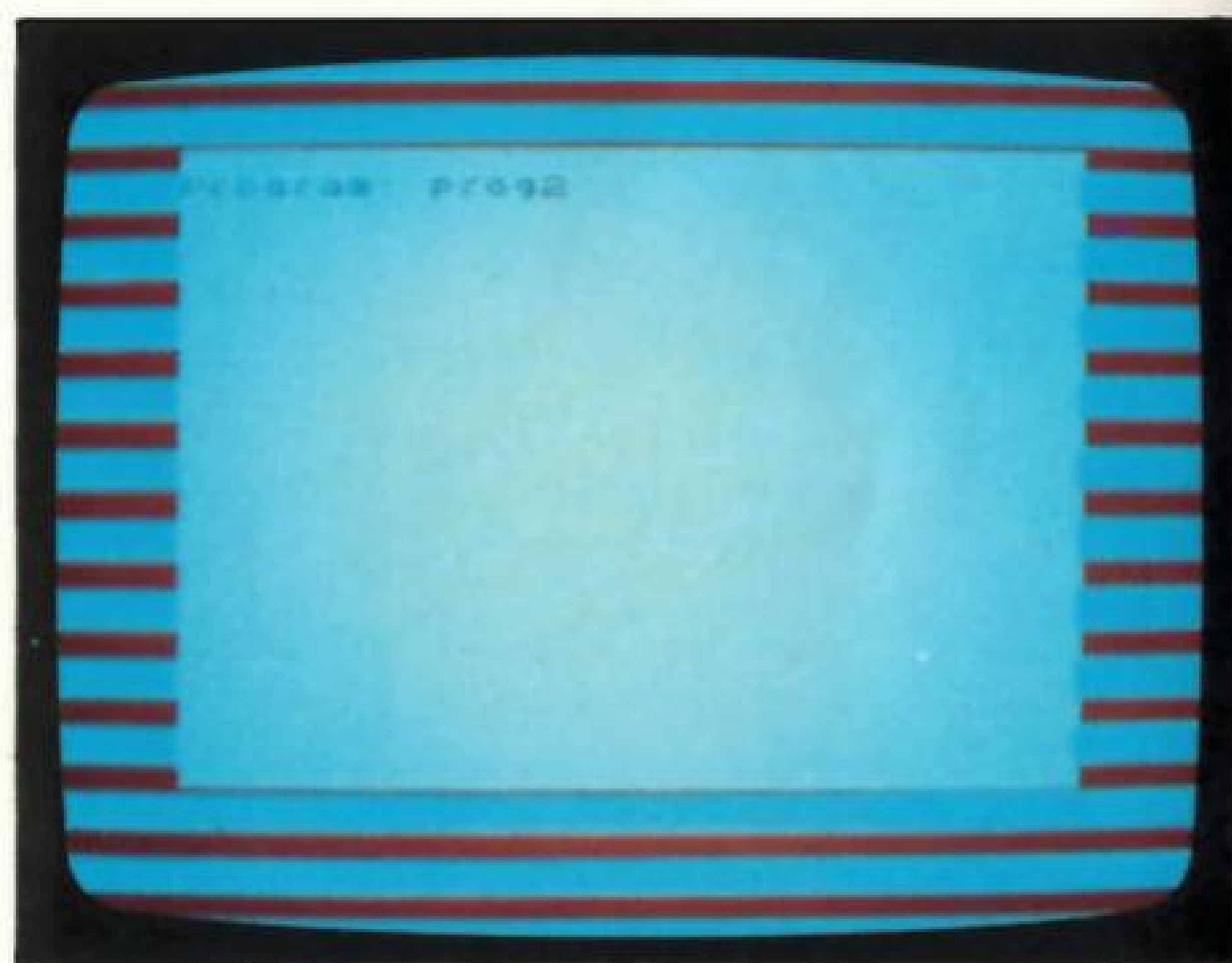
6 על המסך יופיע מרווח קצר ואחריו עוד פסים כחולים ואדומים. זה ילווה בפסים הכחולים והצהובים שנית כאשר הספקטרום שולח את התכנית לסרט. העברת תכנית ארוכה עשוייה לקחת זמן רב למדי.



7 כאשר התכנית נשלחה לסרט, יופיע הדווח 0 OK, 0:1. עצור את הסרט. התכנית שמורה כעת בסרט. אם תרצה ניתן עתה לבדוק זאת או "לאמת".

כיצד תאמת את קיום התכנית

למרות שהמחשב שלח את התכנית לרשמקול הקלטות, אינך יכול להיות בטוח שהתכנית הוקלטה בהצלחה על הסרט. אך הספקטרום מסוגל לבדוק זאת עבורך. נוהל זה נקרא אימות. ראשית הְחזר את הסרט לתחילת התכנית, וחבר את שקע EAR בספקטרום לשקע EAR ברשמקול. (ניתן להשאיר את שקעי MIC מחוברים). כעת הקש VERIFY מלווה בשם התכנית במרכאות. לחץ ENTER והפעל את הרשמקול. על המסך יראה אותו רצף של פסים כחולים ואדומים, ורצועות כחולות וצהובות. שם התכנית יופיע וישאר עד סיום תהליך האימות.



כאשר הקטע השני של כחול וצהוב יופיע כמתואר למטה, ייראה הדווח:

0 OK,0:1

זה אומר שהספקטרום השווה את התכנית שעל הסרט עם התכנית שעדיין נמצאת בזכרון ומצא שהן זהות. התכנית אומתה בתוצאה חיובית.



אתה יכול להיות בטוח כעת שהתכנית נמצאת על הסרט.

עצות לשמירת תוכנה

1. כתוב את שם התכנית על תוית או כרטיס הקלטת כאשר היא מוקלטת. השתמש באותן אותיות גדולות וקטנות המופיעות על המסך. אם יש מונה ברשמקול העזר בו לאיתור התכנית ורשום את מספר המונה על התוית ליד שם התכנית.
2. לפני שתקליט את התכנית, רשום את שם התכנית בתחילת התכנית בעזרת הצהרת REM, לדוגמה:

```
5 REM SPIDER Program Version 3
```

המחשב מתעלם מכל הצהרות REM בזמן ריצת התכנית, וניתן להשתמש ב־REM לצורך הערות ותזכורות בגוף התכנית, לפי הצורך.

אם אינך מקבל דווח זה, סימן שיש תקלה כלשהי. ראשית בדוק בתרשים הזרימה "איתור תקלות טעינת תוכנה" בעמ' 16, מכיוון שיתכן שהתכנית מוקלטת כראוי על הסרט אך אינה נטענת שנית למחשב לצורך אימות. אם התקלה כאן, תקן אותה, גלגל חזרה את הסרט ואמת את התכנית שנית. אם המחשב עדיין לא מאמת את התכנית, עיין בתרשים הזרימה "איתור תקלות איחסון תוכנה" בעמוד הבא. אל תלחץ NEW, RESET או שתכבה את המחשב, כי אז תמחק התכנית מהזכרון ללא קיום עותק אמין על סרט.

הפעלת תכנית אוטומטית

ניתן להקיש SAVE ואחריו את שם התכנית עם LINE 1, לדוגמה:

```
SAVE"SPIDER" LINE 1
```

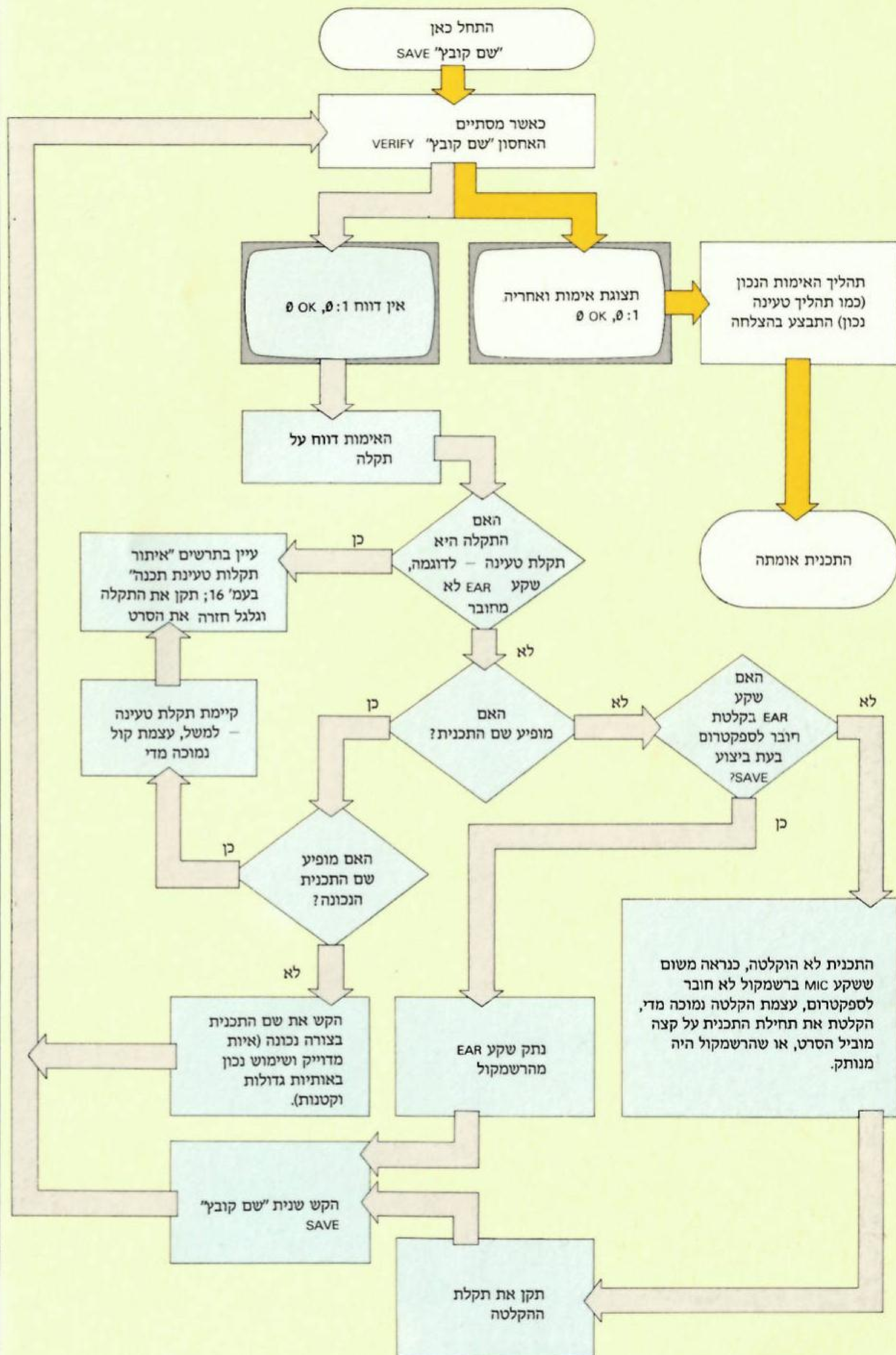
נוהל ההקלטה אינו שונה מקודם אך בזמן האימות, אל תכלול LINE 1 אחרי VERIFY ושם התכנית. תכניות שיוקלטו עם LINE 1 יופעלו אוטומטית בזמן הטעינה. אין צורך להשתמש ב־RUN (אך זכור לעצור את הסרט כאשר התכנית מתחילה). מה שקורה הוא, שהתכנית מתחילה בשורה 1 ואם היא אינה קיימת, המחשב ידלג לשורה הראשונה בתכנית.

שינוי 1 למספר אחר יגרום להפעלה אוטומטית של התכנית בשורה המוקשת.

שמירה על סרט של SCREEN $, CODE ו־DATA.

ניתן להשתמש ב־SAVE יחד עם CODE או SCREEN $ לשמירת קטע מזכרון הספקטרום ועם DATA לאיחסון מערך. ראה את ההסברים ב"מדריך עיון למתכנת".

איתור תקלות איחסון תכנה



למד על הספקטרום +ZX

פרק זה לוקח אותך על תוך קרבי הספקטרום +ZX בהסבירו כיצד פועלים הרכיבים השונים מתחת למקלדת וכיצד הם מתחברים יחד כדי שהמחשב יתפקד. כמו כן מוסבר בפרק כיצד תוכל להשתמש ברכיבים "הקפיים" – התקנים נוספים המאפשרים לשפר את ביצועי הספקטרום ולהביאו לדרגת מערכת מחשב מלאה. לבסוף, תמצא כאן יותר על הצד הטכני במחשב – כולל הצורה בה מאורגן הזכרון, יחד עם המפרט הטכני של הספקטרום.



מה נמצא בפנים?

קרא הלאה במטרה לגלות — ואל תנסה לפתוח את הספקטרום ZX+ כדי למצוא כיצד הוא פועל. בצורה זו רק תפגע בתקפות האחריות ואתה עלול לגרום לנזק רב.

בתוך הקופסה נמצאים שני מחברי כבל שטוח המקשרים את המקלדת לשאר רכיבי הספקטרום. כל יתר הרכיבים הללו מותקנים על כרטיס מעגל מודפס אחד בלבד. הכרטיס נושא רכיבים חשמליים תקניים כגון נגדים וקבלים, אך הפריטים העיקריים הם השבבים הזעירים המרובעים בצבע שחור (ג'וקים), אשר מסודרים ביחידות או בקבוצות.

מה בתוך השבב

החלק התפקודי של שבב זעיר הוא בעצם קטן בהרבה מהמארז הפלסטי הנושא אותו. המארז מתוכנן בעיקר לתמוך בכל החיבורים שהשבב דורש, ומאפשר לו להתקע בשקעים הנמצאים בכרטיס המעגל. השבב עצמו הוא פיסת צורן דקה אשר מכילה אלפים רבים של צמתים חשמליים. כל צומת פועלת כמתג לצורך עצירה, העברה או אחסון של אותות חשמליים המגיעים אליה. למרות שזהו בעצם תהליך פשוט, קיימים צמתים כה רבים הפועלים יחד עד כי ניתן להפיק אותות המאחסנים או מעבדים מידע במהירות עצומה ודיוק רב. הספקטרום ZX+ כולל מספר שבבים שונים, כשכל אחד מתוכנן לבצע משימה מסוימת לצורך ריצת המחשב.

כיצד מתחברים השבבים?

בסך הכל, הספקטרום הוא מעגל חשמלי מסובך ביותר. אותות קוד הכוללים דחפים חשמליים חולפים ללא הרף בנתיבים שבתוך ובין השבבים והרכיבים כדי שהמחשב יפעל.

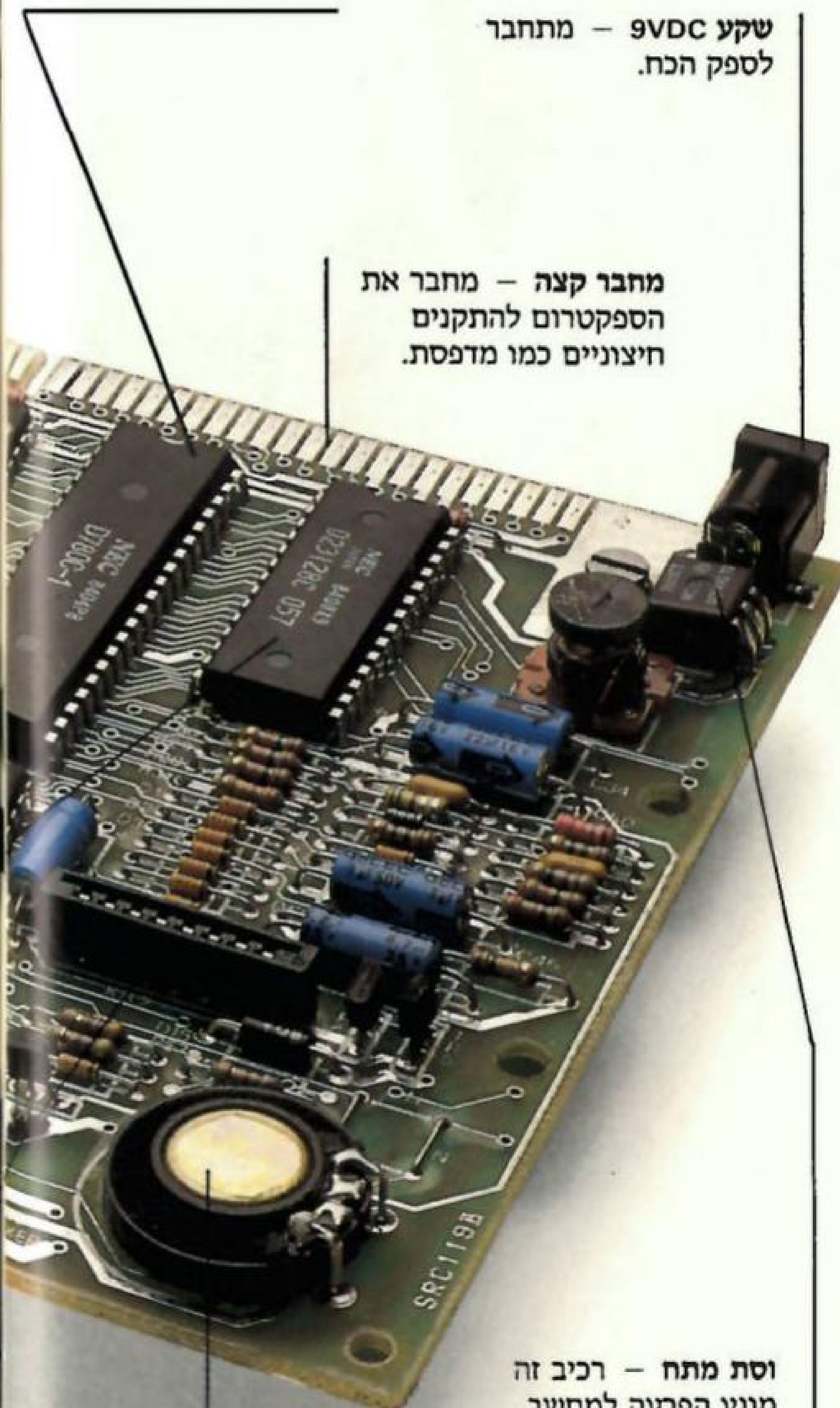
כיצד נשמר כל דבר בסדר הנכון כך שהאות הנכון יגיע למקום הנכון בזמן הנכון? בתוך אחד השבבים טמון שעון המחשב. שעון זה פועל ע"י פליטת דחפים חשמליים — 3.5 מליון דחפים בשניה. דחפים אלה נעים בצורה סדירה דרך המעגלים ליצירת אותות קוד אשר מבקרים את פעולות החלקים ושומרים על הסדר הנכון.

קרבי הספקטרום ZX+
במבט זה של כרטיס מעגל הספקטרום ZX+ הוסרו שני מחברי כבל שטוח של המקלדת. כאשר הספקטרום נמצא בשימוש, כל לחיצה על מקש מביאה למגע בין צמד חוטים מתחת למקלדת. מגע זה שולח את קוד ליע"מ.

יחידת עיבוד מרכזית (CPU) — ה"מוח" של המחשב, היע"מ הוא מעבד זעיר מדגם Z80. המעבד מבצע את כל פעולות החישוב ומבקר את הפעולה הכללית של הספקטרום.

שקע 9VDC — מתחבר לספק הכח.

מחבר קצה — מחבר את הספקטרום להתקנים חיצוניים כמו מדפסת.



וסת מתח — רכיב זה מונע הפרעה למחשב כתוצאה משינויים כלשהם במתח האספקה.

רמקול — מפיק צלילים במידת הצורך.

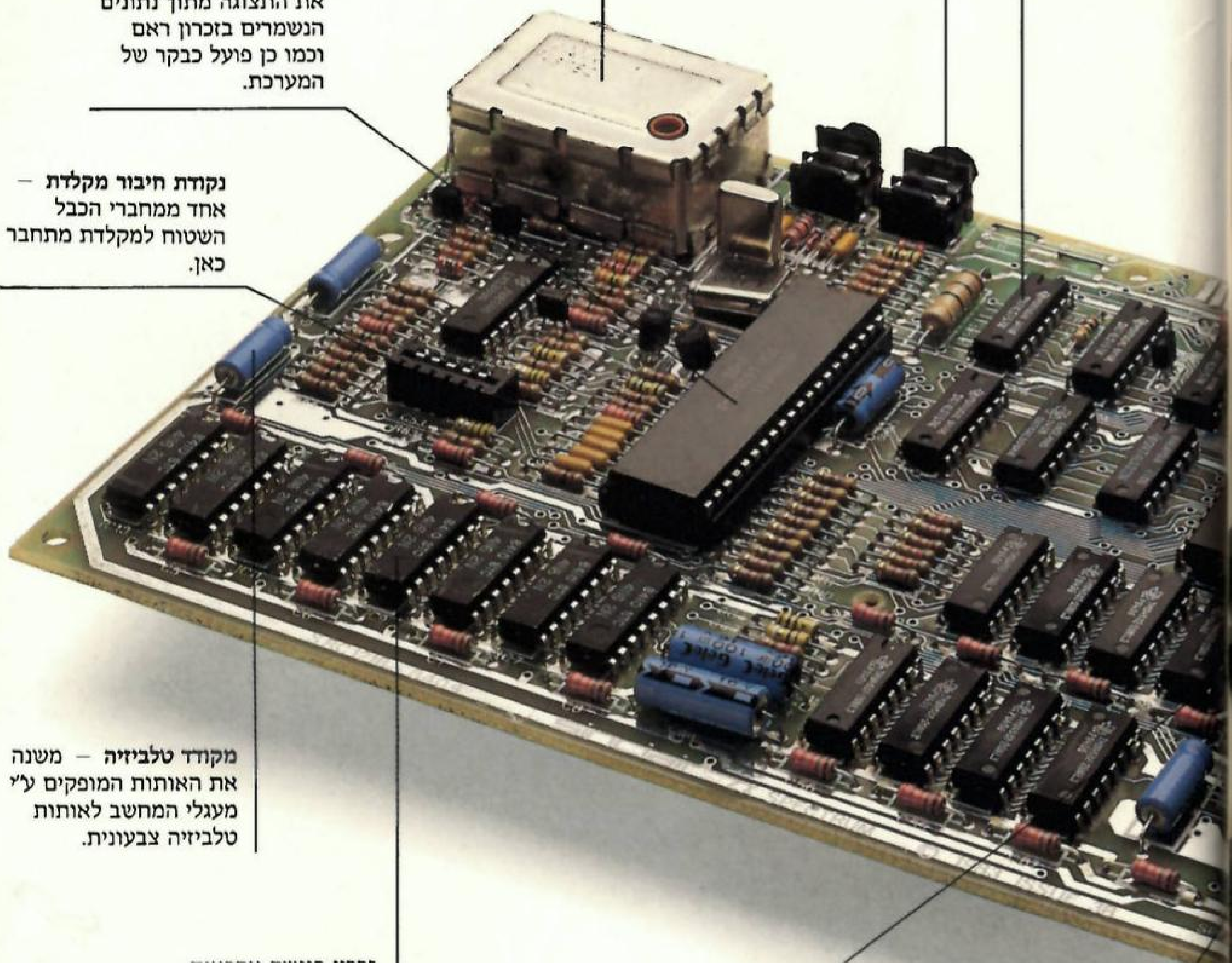
שקעי קלטת – אלה משמשים להעברת נתונים ותכניות מהזכרון לסרט, ולהזנתם חזרה מסרט לזכרון המחשב.

שבבים לוגיים – שבבים אלה פועלים כממשק בעת החלפת מידע בין היע"מ והזכרון בגישה אקראית (ראם).

תפוקת טלביזיה – רכיב זה מפיק את האותות הנשלחים למקלט הטלביזיה.

מערך לוגי בלתי תלוי (ULA) – שבב זה מפיק את התצוגה מתוך נתונים הנשמרים בזכרון ראם וכמו כן פועל כבקר של המערכת.

נקודת חיבור מקלדת – אחד ממחברי הכבל השטוח למקלדת מתחבר כאן.



מקודד טלביזיה – משנה את האותות המופקים ע"י מעגלי המחשב לאותות טלביזיה צבעונית.

זכרון בגישה אקראית (RAM) – שבבים אלה מכילים את התכנית המוזנת למחשב וכל נתון מסויים הנדרש ע"י התכנית, כגון ערכים המיוצגים ע"י משתנים. תכולת 48 ק' של זכרון ראם ניתנת לשינוי מן המקלדת, וניתן גם למחקה כליל ע"י כבוי המחשב או לחיצת RESET.

זכרון קריאה בלבד (ROM) קטע מהזכרון בן 16 ק' המכיל את הוראות ההפעלה הקבועות הנדרשות ע"י היע"מ. בין היתר, הוראות אלה ממירות את תכניות הבייסיק לשפה שהיע"מ מבין. תכולת שבבי זכרון אלה לא ניתנת לשינוי מן המקלדת.

נקודת חיבור מקלדת – אחד ממחברי הכבל השטוח למקלדת מתחבר כאן.

כיצד פועל הספקטרום +ZX ?

הספקטרום +ZX, בדומה למערכות מיקרו מחשב אחרות בנוי מארבעה חלקים עקריים. אלה הם יחידות הקלט, כגון מקלדת, אשר מזינה נתונים או תכנית אל המחשב; הזכרון הארעי והזכרון הקבוע, אשר אוגרים נתונים, תכניות והוראות הפעלה; יחידת העיבוד המרכזית CPU, אשר מפעילה את פקודות התכנית על הנתונים, ויחידות הפלט, אשר נותנות את התוצאה.

הכנסה והרצת תכנית

מה קורה בתוך הספקטרום כאשר אתה מקיש ומריץ תכנית פשוטה? הנה למשל דוגמה בת שורה אחת:

```
10 PRINT 6 + 2
```

ראשית, אתה מפעיל את המקלדת. מתחת למקשים נמצאת רשת חוטים צולבת. בכל פעם שנלחץ מקש, נוצר מגע בין זוג חוטים השולח אות קוד אל היע"מ. היע"מ בתורו שולח את הקוד לזכרון ראם, בו הוא מאוחסן.

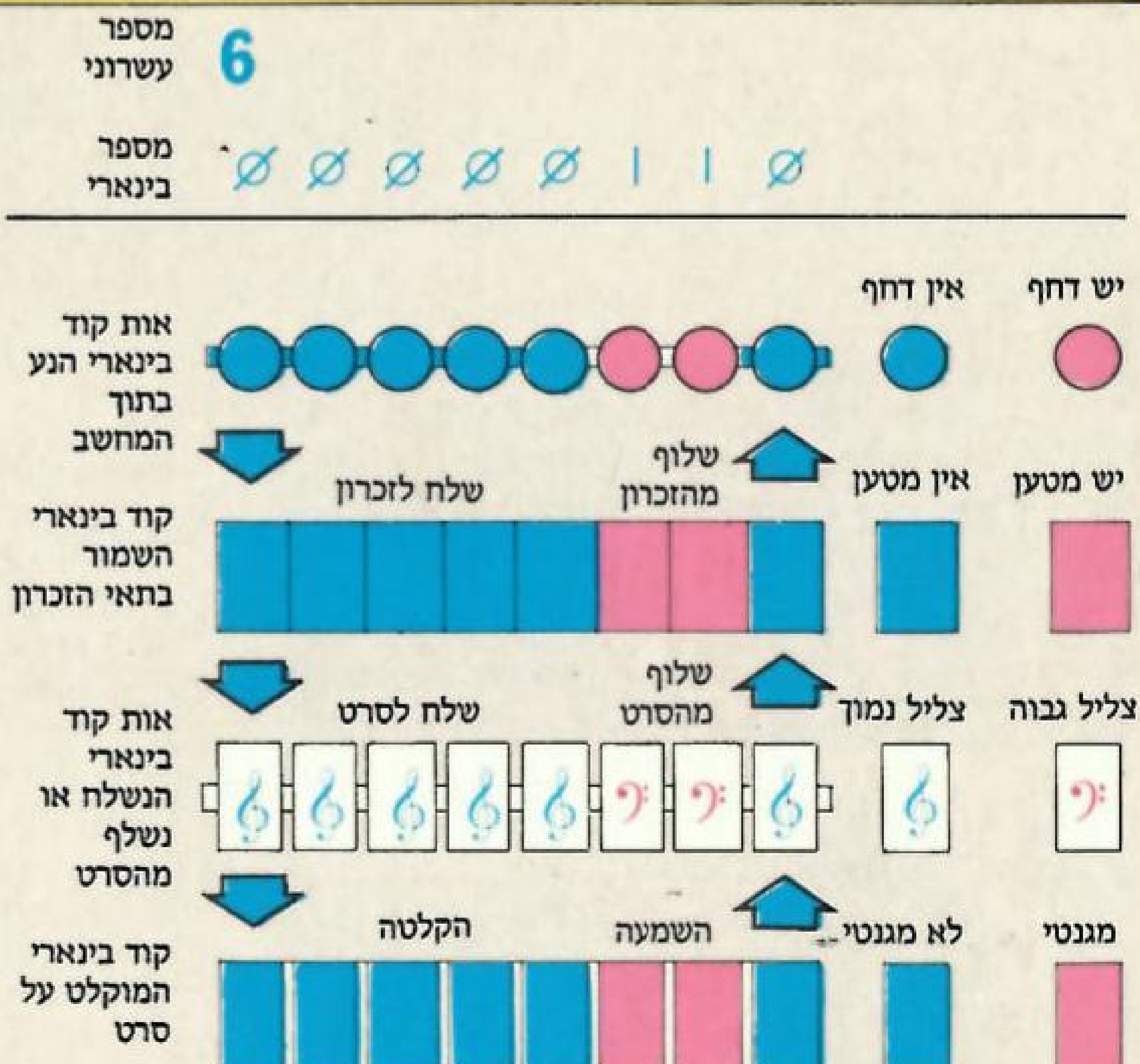
כאשר אתה מריץ את התכנית, היע"מ לוקח את האותות המאוחסנים בראם בזה אחר זה, לפי סדר התכנית. בתחילה הוא מקבל את הקוד עבור PRINT, אשר מורה לו לקחת קוד פעולה מסויים מזכרון רום. קוד פעולה זה נשלח ליע"מ אשר מתכונן לבצע את הפעולות הנדרשות להצגת ערך על המסך. כעת היע"מ מקבל את הערך של 6 מן הראם. גם ערך זה נשלח בצורת קוד, והיע"מ מאחסן אותו בזכרון פנימי קטן הנקרא אוגר. בהמשך מגיע הקוד עבור פעולת החיבור, ושוב היע"מ מקבל את קוד הפעולה הנדרש מזכרון רום. לבסוף, היע"מ לוקח מזכרון ראם את הקוד עבור 2. הוא מחבר קוד זה לערך הנמצא באוגר לקבלת התוצאה (8). כעת היע"מ ממיר את התוצאה למערך קודים אחר ושולח אותו לקובץ התצוגה. זהו החלק בזכרון ראם אשר מכיל קודים לכל מה שתראה על המסך, ואז המספר 8 מופיע על המסך.

איחסון תכנית

אם תדרוש מהספקטרום לשמור את התכנית על סרט, היע"מ יפנה שוב לזכרון ראם לקבלת הקודים. אך במקום לפעול עליהם, הוא שולח את הקודים ליחידת המרה אשר משנה אותם לאותות קול. אותות אלה נשלחים לרשמקול ומוקלטים על סרט.

כאשר תטען מאוחר יותר את התכנית, אותות הקול מהרשמקול ישתנו חזרה לקודי מחשב ע"י יחידת ההמרה. היע"מ ישלח אותם חזרה לזכרון ראם, שם הם יאוחסנו עד שיידרשו.

קודים בינאריים



כל הקודים הגורמים לספקטרום לפעול הם בצורה בינארית. הם נקראים בינאריים מפני שכולם מורכבים משני סוגי אותות בלבד. הם ניתנים לייצוג כמספרים בינאריים, כלומר, מספרים המכילים רק שתי ספרות — Ø ו־1. המספר הבינארי עבור 6, למשל, הוא ØØØØØ11Ø.

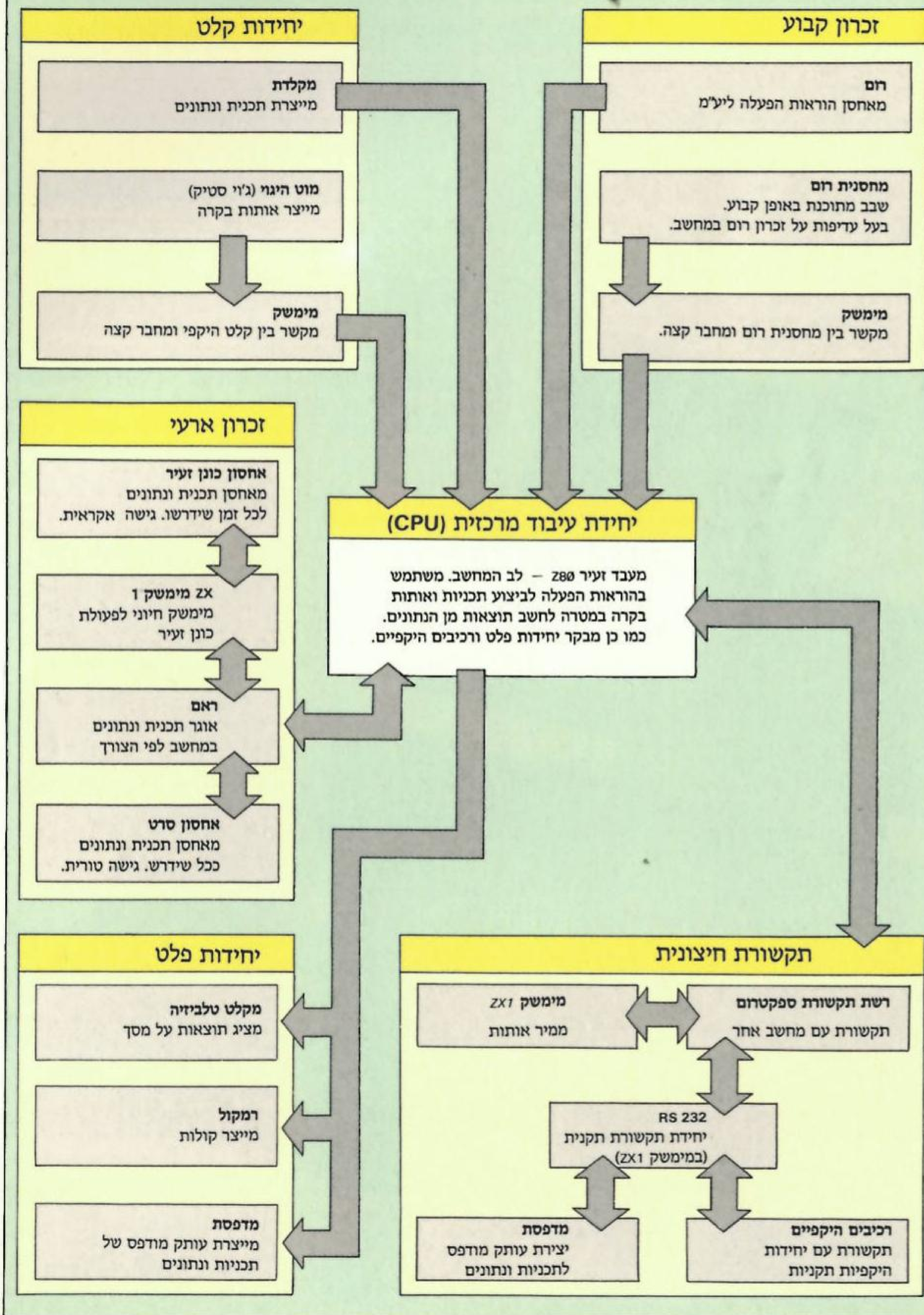
בתוך הספקטרום, הקודים בנויים מרצף של דחפים חשמליים מהירים. אם מגיע דחף חשמלי לנקודה מסויימת, הוא מייצג 1 בינארי. אם דחף לא מגיע תוך פרק זמן מסויים, זה מייצג Ø בינארי. בקוד מחשב, 6 הוא *לא-כן-כן-לא-לא-לא-לא-לא*. הספקטרום מטפל ב־8 קודים בו זמנית.

בתרשים זה תוכל לראות כיצד המחשב משתמש בסוגים שונים של קידוד בינארי להעברת נתונים ממקום אחד לשני.

נתיבי קלט-פלט בספקטרום

תרשים זה מתאר כיצד עובר מידע מקודד מיחידות קלט כמו המקלדת, דרך מערכת העיבוד של הספקטרום, ומשם ליחידות פלט כמו מסך הטלביזיה.

חיצים חד־כווניים מציינים נתיבים הפועלים בכוון אחד בלבד. חיצים דרכווניים מציינים נתיבים הפועלים בשני הכוונים.



כיצד לחבר רכיבים היקפיים

ניתן לשפר את ביצועי הספקטרום +ZX ולהגיע למערכת מחשב מושלמת ורבת עצמה תוך שימוש ברכיבים היקפיים של סינקלייר ותואמי ספקטרום אחרים. הרכיב המרכזי במערכת הוא מימשק ZX1, המאפשר לך לחבר כוננים זעירים לטיפול מהיר ונוח בתכניות ונתונים, ומתחבר גם למגוון רחב של רכיבים היקפיים שונים, כולל מחשבי ספקטרום אחרים. בעזרת מימשק זה, תוכל לחבר את הספקטרום למדפסות תקניות, ולחבר את המחשב למודם המאפשר שידור וקליטת תכניות ונתונים דרך קוי טלפון. קיימים גם מימשקים אחרים המקשרים מחסניות רום נתקעות למחשב לצורך טעינת תכניות מיידית. בעזרת אלה, תוכל גם לחבר מוט היגוי כדי להשיג בקרה נוחה על משחקי מחשב.

מחבר קצה – רכיבים היקפיים מתחברים למחשב דרך מחבר זה.

אזהרה – רכיבים היקפיים חייבים להיות מחוברים לפני הכנסת מתח למחשב.

מדפסות תואמות-ספקטרום

מדפסות מסויימות מתחברות ישירות למחבר הקצה בספקטרום. אם יש כבר ברשותך מדפסת סינקלייר ZX, למשל, תוכל לחברה למחשב ללא שימוש במימשק. סוג זה של מדפסת מתחבר גם לגב המימשק ZX1. ברם, כדי להשתמש במדפסות הדורשות פלט RS232, עליך להתחבר לשקע D במימשק ZX1.

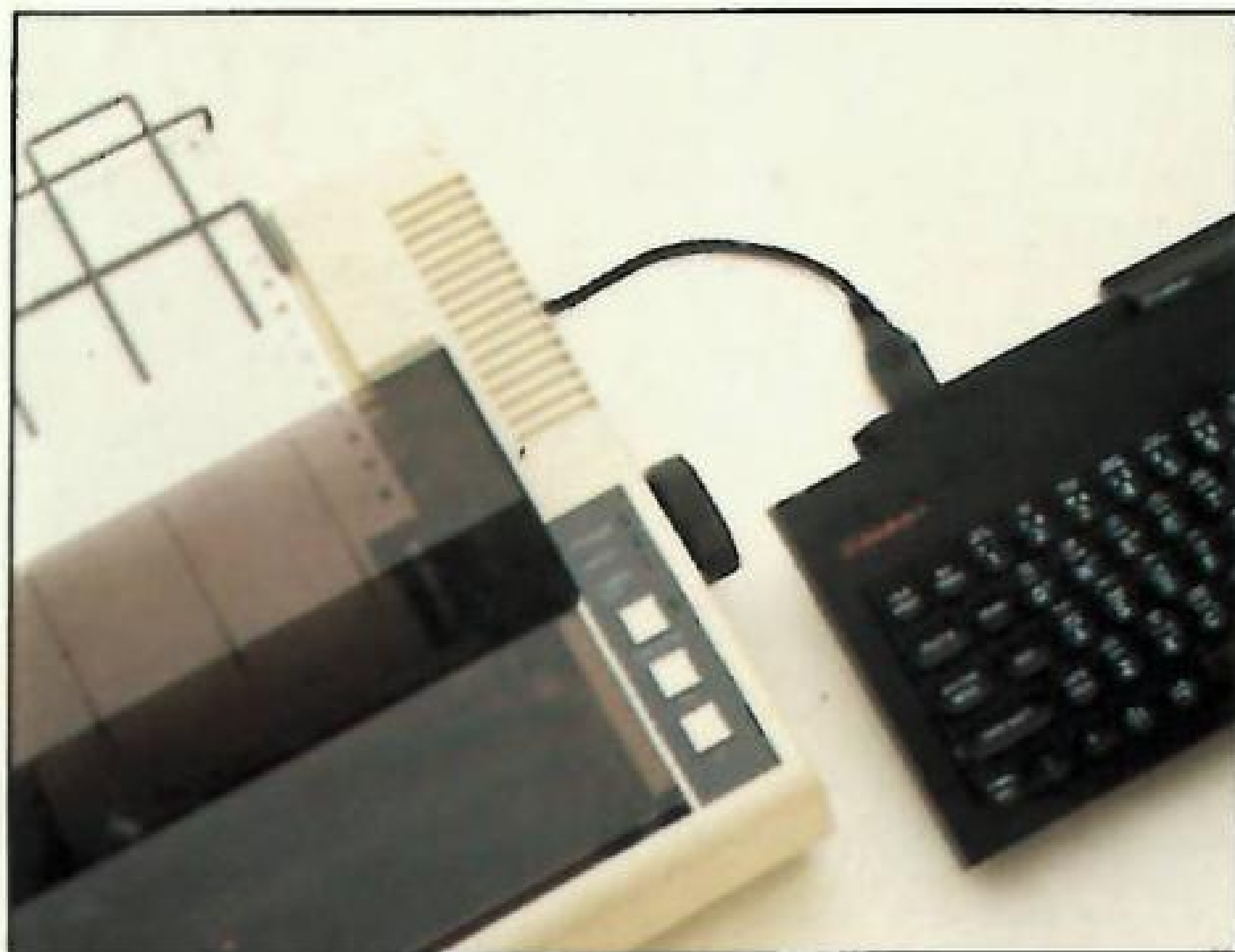
מימשק תקליטון רום/מוט היגוי

מימשק ZX1

יחידת מימשק ZX1 מתחברת לבסיס ולחלק האחורי של הספקטרום. היחידה מקשרת את הספקטרום לעד 8 כוננים זעירים, עד 63 מחשבי ספקטרום אחרים, ובעזרת יחידת המימשק התקני RS232, למגוון רחב של רכיבים היקפיים תקניים.

כוננים זעירים ותקליטונים מחליפים את הרשמקול והסרטים לצורך אחסון תכניות ונתונים. על ידי הכנסת תקליטוני כונן זעיר תוכל לשמור, לאמת ולטעון תכניות תוך שניות. כל תקליטון יכול לאחסן עד 85 ק' נתונים ובעזרת השימוש במקסימום של 8 כוננים זעירים, יהיה לספקטרום קיבול מאגר מקוון (On-line) של עד 680 ק'! כל תכנית מאותרת באופן אוטומטי עם זמן גישה טיפוסי של 3.5 שניות. בעזרת כבל רשת התקשורת המסופק עם יחידת המימשק, תוכל לקשר את המחשב עם מחשב אחר — ספקטרום ZX או מחשב ספקטרום +ZX נוסף. רשת תקשורת זו ניתנת להרחבה עד 63 ספקטרומים אחרים. הנתונים מועברים ביניהם בקצב של 10000 תוים בשנייה.

יחידת מימשק ZX1 כוללת גם מימשק RS232 עם שקע D בן 9 מגעים אשר דרכו ניתן לחבר לספקטרום מדפסות, רכיבים היקפיים תקניים אחרים, מודמים ומחשבים אחרים. כמו כן קיים גם כבל מימשק תקני.



מחסניות רום ומוטות היגוי

מימשקים כמו מימשק ZX2 מאפשרים חיבור מחסניות רום ומוטות היגוי. מחסניות רום נטענות מייד עם ההפעלה כדי לטעון במהירות תכניות בעלות זמן טעינה גדול מסרט.

חיבור רכיבים היקפיים — בעת הפעולה, מימשק ZX1 נתקע במחבר הקצה כך שהוא נמצא מאחורי ומתחת למחשב. האיור כאן מראה את המערכת לפני חיבור המחשב למימשק.



טעינת כונן זעיר — תקליטוני כונן זעיר מוכנסים לחריץ בחזית הכונן.



חיבור מחסנית רום — המחסנית מוכנסת לשקע המימשק. כאשר המחשב מופעל, התכנית נטענת אוטומטית, ועוקפת את זכרון רום במחשב.



כבל שטוח — הכבל השטוח מחבר את הכונן הזעיר למחשב דרך יחידת מימשק ZX1.

יחידות כונן זעיר — ניתן לחבר עד 8 יחידות איחסון כאלה לספקטרום אחד.

הערה — לספקטרום +ZX יש שתי רגליים פנימיות לצורך הטיית המקלדת. אין להשתמש ברגליים אלה כאשר מורכב מימשק ZX1.

מיפוי זכרון בספקטרום +ZX

אם תתבונן בתצלום של קרבי הספקטרום בעמ' 42-43, תראה שקיים שבב רום אחד, ו־16 שבבי ראם קטנים יותר. שבבים אלה מהווים את זכרון הספקטרום. הזכרון מורכב מ־65536 יחידות אחסון שכל אחת מהן מכילה בית אחד (מספר בתחום שבין Ø ועד 255). כל יחידה מזוהה על ידי מספר הנקרא כתובת.

רום הוא קיצור מאנגלית עבור זכרון קריאה בלבד. זהו חלק מהזכרון אשר מכיל הוראות הפעלה עבור יחידת העיבוד המרכזית. זהו רום 16 ק', שפירושו הוא זכרון בתכולה של 16×1Ø24 (16384) בתים או כתובות. הבתים ניתנים לקריאה בלבד מזכרון זה, כך שאינם ניתנים לשינוי. (אילו היו ניתנים לשינוי, המחשב היה מפסיק לפעול.) אתה יכול לבחון את הבית בכל כתובת בעזרת פקודת PEEK.

ראם פירושו זכרון בגישה אקראית, והוא מכיל את התכניות והנתונים המוזנים למחשב. לספקטרום יש ראם 48 ק', כלומר, הזכרון מכיל 48×1Ø24 (49152) בתים או כתובות. גישה אקראית פירושה, שבית בכל כתובת שהיא ניתן לשינוי, והדבר ניתן לביצוע בעזרת הפקודה POKE.

כתובות הזכרון משתרעות מ־Ø עד 65535, כשהרבע הראשון עבור זכרון רום והיתר לזכרון ראם.

משתני מערכת

בעמודה שלהלן מתואר ארגון הזכרון בספקטרום. בתרשים זה תוכל לראות היכן נמצאים החלקים השונים אשר מבקרים את המחשב. כמה מאלה עשויים לשנות מיקום, וגבולותיהם נתונים בתחום של משתני מערכת.

משתני המערכת בספקטרום אינם משתנים כמו אלה הקיימים בבייסיק. אלה הם פשוט שמות לערכים שימושיים מסויימים הנמצאים בכתובות או מיקומים מסויימים בזכרון. מטרת השם היא לעזור לך לזכור את משמעות הערך המסויים השמור במקום זה. לדוגמה, משתנה המערכת RAMTOP הוא הכתובת העליונה בזכרון ראם. תחום זכרון זה מכיל תכנית בייסיק ואת ערכי משתניה. כתובתו של RAMTOP היא 2373Ø.

מפת זכרון

אזור	כתובת / משתנה מערכת
גרפיקה בהגדרת־משתמש	
	RAMTOP
מחסנית GOSUB	
פנוי	
	STKEND
מחסנית מחשבון	
	STKBOT
שטח עבודה זמני	
נתוני קלט	
	WORKSP
פקודה או שורה העוברת עריכה	
	E-LINE
משתנים	
	VARS
תכנית בייסיק	
	PROG
נתוני ערוץ	
	CHANS
מיפוי כונן זעיר	
	23734
משתני מערכת	
	23552
מאגר מדפסת	
	23296
מאפיינים	
	22528
קובץ תצוגה	
	16384
רום 16 ק'	

ראם 48 ק'

למד על הבייסיק של סינקלייר

פרק זה מתאר את הבייסיק של סינקלייר במלואו. כאן תמצא תקציר של אופן פעולת כל מילת מפתח, ופרטים נוספים על פעולת הבייסיק של סינקלייר. המידע הניתן כאן משתרע ממה שנדרש לצורך תכנית בייסיק הפשוטה ביותר ועד למתקדמת ביותר. פרק זה אינו מיועד לקריאה שוטפת מתחילתו ועד סופו. למעשה, זהו מילון למתכנת אשר מאפשר לך לנצל את מלוא יכולתו של הספקטרום.



מדריך עיון למילות מפתח עבור מתכנת בייסיק של סינקלייר

מדריך זה מכיל תאורים מפורטים של כל מילות המפתח בבייסיק אשר קיימות בספקטרום +ZX. כל רשומה כוללת

- מיקום מילת המפתח
- סוג מילת המפתח
- מטרת מילת המפתח
- השימוש במילת המפתח
- פורמט תיכנות

הפרטים הניתנים עבור מיקום, מטרה ושימוש מסבירים את עצמם. סוג ופורמט כוללים הגדרות מסובכות יותר, וכדי להפיק את מירב התועלת ממדריך זה, עליך לקרוא תחילה בעיון את המידע בעמוד זה.

סוגי מילות מפתח

מילות מפתח נופלות בתחום אחד או יותר מארבעת הסוגים הבאים.

פקודה
מילת מפתח הגורמת לביצוע פעולה וניתנת לשימוש לביצוע פקודה ישירה. פקודה מתבצעת מייד לאחר הקשתה. דוגמאות — RUN, LOAD.

הצהרה
מילת מפתח הגורמת לביצוע פעולה וניתנת לשימוש בשורת תכנית. היא מתבצעת רק כאשר התכנית מורצת. דוגמאות — DRAW, INPUT.

פונקציה
מילת מפתח היוצרת ערך מסוג כלשהו. היא יוצרת חלק מפקודה או הצהרה. דוגמאות — RND, INT.

אוערטור לוגי
מילת מפתח המשמשת לביטוי לוגיקה בהצהרה או פקודה. האופרטור יכול לקבוע או לשנות את אמיתות תנאים מסויימים. בספקטרום יש שלשה אופרטורים לוגיים — AND, OR ו־NOT.

מספרים ומשתנים

מספרים
מאוחסנים בדיוק של עד 9 או 10 ספרות. טווח הטיפול במספרים הוא מ־10^{38} ועד לכ־10^{39} * 4.

משתנים תקפים
משתנים מספריים בכל אורך, המתחילים באות. רווחים לא נחשבים וכל האותיות מומרות לאותיות קטנות. אין הבחנה בין אותיות גדולות וקטנות.

משתני מחרוזת עם אות בודדת כלשהי, בלווית סימן $. אין הבחנה בין אותיות גדולות וקטנות.

משתני מערכים, ראה הסבר בנושא DIM, הדן גם במשתני מערך ותויות.

פורמט מילת מפתח

פורמט מילת המפתח מתאר את התחביר של כל מילת מפתח — כלומר, הצירוף הנכון של מילת המפתח וגורמים אחרים כמו ערכים ומשתנים. הקיצורים הבאים משמשים לצורך הגדרת הפורמט.

קיצור	הסבר	דוגמה
num-const	קבוע מספרי (מספר)	24.5
num-var	משתנה מספרי (משתנה העשוי להכיל מספר)	sum
num-expr	בטוי מספרי (כל צירוף תקף של קבועים מספריים, משתנים ומילות מפתח הנותנים מספר).	sum•24.5 RND•7
int-num-const int-num-var int-num-expr	קבוע, משתנה או בטוי מספרי אשר ערכו מעוגל לשלם הקרוב ביותר.	
string-const	קבוע מחרוזת או מחרוזת (כל צירוף של תוים בין סימני מרכאות).	"ZX Spectrum + "
string-var	משתנה מחרוזת (משתנה העשוי להכיל מחרוזת).	a$
string-expr	בטוי מחרוזת (כל צירוף תקף של קבועי, משתני ומילות מפתח מחרוזת, הנותנים מחרוזת).	a$ + "ZX Spectrum + " a$ (6 TO 8)
letter	כל אות קטנה או גדולה	Y x
letter	כל אות קטנה או גדולה המלווה בסימן $	B$ a$
cond	תנאי או תת־תנאי בתוך תנאי	x=10 AND t<10
statement	כל הצהרת בייסיק אשר תקפה כאשר יש שימוש בה עם הצהרה אחרת.	IF t>10 THEN STOP PRINT INK 2;x
	פרט רשות (אופציה) העשוי לחזור על עצמו.	[ ]

'**הערה:** המונחים ערך **מספרי** וערך מחרוזת מופיעים בטקסט עבור פריטים מספריים או פריטי מחרוזת בהתאמה.

סימנים בבייסיק של סינקלייר

סימן	מיקום	פעולה/שימוש
$	4	משתנה מחרוזת
'	7	התחלת שורה חדשה
(	8	פתח סוגריים
)	9	סגור סוגריים
<=	Q	קטן מ... או שווה ל...
<>	W	לא שווה ל...
>=	E	גדול מ... או שווה ל...
<	R	קטן מ...
>	T	גדול מ...
↑	H	העלה בחזקה
−	J	חיסור / שלילי
+	K	חיבור / חיובי / שרשור מחרוזות
=	L	שווה ל...
:	Z	הפרדת הצהרות בשורת תכנית
/	V	חילוק
*	B	כפל
.	מקש נפרד	נקודה עשרונית
;	מקש נפרד	מציג בעמודה הבאה מפריד בין הצהרות בתוך הצהרת תכנית
"	מקש נפרד	פתיחת וסגירת מחרוזות
,	מקש נפרד	מציג בעמודה Ø או 16 הפרדת ערכים המלווים מילות מפתח

מערך התווים בספקטרום +ZX

	0	1	2	3	4	5	6	7	8	9
0					TRUE VIDEO	INV VIDEO	PRINT comma	EDIT	cursor left	cursor right
10	cursor down	cursor up	DELETE	ENTER	number	GRAPHICS MODE	INK control	PAPER control	FLASH control	BRIGHT control
20	INVERSE control	OVER control	AT control	TAB control						
30			space	!	"	#	$	%	&	'
40	(	)	*	+	,	−	.	/	Ø	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	/	]	↑	_	£	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~	©	□	▝
130	▘	▀	▗	▐	▚	▜	▖	▞	▌	▛
140	▄	▟	▙	█	GRAPHICS A	GRAPHICS B	GRAPHICS C	GRAPHICS D	GRAPHICS E	GRAPHICS F
150	GRAPHICS G	GRAPHICS H	GRAPHICS I	GRAPHICS J	GRAPHICS K	GRAPHICS L	GRAPHICS M	GRAPHICS N	GRAPHICS O	GRAPHICS P
160	GRAPHICS Q	GRAPHICS R	GRAPHICS S	GRAPHICS T	GRAPHICS U	RND	INKEYS	PI	FN	POINT
170	SCREENS	ATTR	AT	TAB	VALS	CODE	VAL	LEN	SIN	COS
180	TAN	ASN	ACS	ATN	LN	EXP	INT	SQR	SGN	ABS
190	PEEK	IN	USR	STRS	CHRS	NOT	BIN	OR	AND	<=
200	>=	<>	LINE	THEN	TO	STEP	DEF FN	CAT	FORMAT	MOVE
210	ERASE	OPEN #	CLOSE #	MERGE	VERIFY	BEEP	CIRCLE	INK	PAPER	FLASH
220	BRIGHT	INVERSE	OVER	OUT	LPRINT	LLIST	STOP	READ	DATA	RESTORE
230	NEW	BORDER	CON-TINUE	DIM	REM	FOR	GO TO	GO SUB	INPUT	LOAD
240	LIST	LET	PAUSE	NEXT	POKE	PRINT	PLOT	RUN	SAVE	RANDOM-IZE
250	IF	CLS	DRAW	CLEAR	RETURN	COPY				

ABS ערך מוחלט

מיקום במקלדת

EXTEND MODE
G

פונקציה

ABS נותנת את הגודל המוחלט של ערך מספרי, כלומר, הערך ללא סימן חיובי או שלילי.

כיצד להשתמש ב־ABS

ABS מלווה בערך מספרי. בטוי חייב להיות בתוך סוגריים, לדוגמה:

```
5Ø LET x = ABS(y - z)
```

ABS מחזירה את הערך המוחלט של ערך מספרי.

דוגמה

הפקודה

```
PRINT ABS — 34.2
```

מציגה 34.2.

פורמט

ABS num-const
ABS num-var
ABS (num-expr)

ACS ארק קוסינוס

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT W

פונקציה

ACS מחשבת ערך זוית מתוך הקוסינוס שלה.

כיצד להשתמש ב־ACS

ACS מלווה בערך מספרי. בטוי חייב להיות בתוך סוגריים, לדוגמה:

```
6Ø LET x = ACS (y * z)
```

הערך המלווה את ACS (y * z) שלעיל הוא הקוסינוס של הזוית הנדרשת, הנמצא בתחום 1 — עד 1. ACS מחזירה את ערך הזוית ברדיאנים. כדי להמיר רדיאנים למעלות, הכפל את הערך המוחזר ע"י ACS ב־180/PI.

דוגמה

הפקודה

```
PRINT 18Ø/PI * ACS Ø.5
```

מציגה 6Ø, הזוית במעלות שיש לה קוסינוס של Ø.5.

פורמט

ACS num-const
ACS num-var
ACS (num-expr)

AND

מיקום במקלדת

SYMBOL SHIFT Y

אופרטור לוגי/פונקציה

AND פועלת כאופרטור לוגי לבדיקת אמיתות הצירוף של תנאים. רק אם **כל** התנאים הם אמת סה"כ הצירוף הוא אמת. AND משמשת גם כפונקציה לביצוע פעולות בינאריות על שני ערכי מספרים או מחרוזות.

כיצד להשתמש ב־AND

בתור אופרטור לוגי, AND מקשרת שני תנאים בהצהרה בה יש לבחון את קיום כל התנאים, לדוגמה

```
9Ø IF x = y + z AND time < 1Ø
  THEN PRINT "Correct"
```

רק אם שני התנאים (x = y + z ו־time <1Ø) הם אמת, המחשב יציג Correct. אם אחד או שני התנאים שקרי, אזי כל הצירוף שקרי, ובמקרה זה התכנית תמשיך לשורה הבאה.

AND כפונקציה

בתור פונקציה, AND יכולה לפעול על שני ערכים מספריים, לדוגמה

```
5Ø LET x = y AND z
```

AND מחזירה את הערך הראשון (y) אם השני (z) לא שווה ל־Ø, ומחזירה Ø אם הערך השני (z) הוא Ø.

AND יכולה לפעול גם על משתנה מחרוזת בתנאי שהוא בא לפני AND. אחרי AND חייב תמיד לבוא ערך מספרי, לדוגמה

```
5Ø LET a$ = b$ AND z
```

AND מחזירה את הערך הראשון (b$) אם השני (z) אינו אפס, ומחרוזת ריקה ("") אם הערך השני (z) הוא Ø.

שים לב לכך שהספקטרום +ZX מקצה ערך של 1 לתנאי אמת ו־Ø לתנאי שקרי, ומכיר כל ערך שאינו אפס כאמת ו־Ø כשקר. הוא לא מעריך צירופים של ערכים מספריים בהתאם לטבלות אמת תקניות.

דוגמאות

```
6Ø LET correct = (x = y + z) AND
  time < 1Ø
7Ø LET score = score + 1Ø * (1
  AND correct)
8Ø LET a$ = ("Out Of Time Or Not"
  AND NOT correct) + "Correct"
```

אם שני התנאים בשורה 6Ø הם אמת, אזי המשתנה המספרי correct מקבל ערך של 1. ואז score מוגדל ב־1Ø ו־a$ הופך "Correct". אם אחד משני התנאים שקרי, אזי correct מקבל ערך של Ø; Score לא משתנה ו־a$ הופך ל־"Out Of Time Or Not Correct"

פורמט

Cond AND cond
num-expr AND num-expr
string-expr AND num-expr

ASN ארק סינוס

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT Q

פונקציה

ASN מחשבת את ערך הזוית מתוך הסינוס.

כיצד להשתמש ב־ASN

ASN מלווה בערך מספרי. בטוי חייב להיות בין סוגריים, לדוגמה

```
6Ø LET x = ASN (y * z)
```

הערך המלווה את ASN (y * z) שלעיל הוא סינוס הזוית הנדרשת ונמצא בתחום שבין 1 — ועד 1. ASN מחזירה את ערך הזוית ברדיאנים. כדי להפוך רדיאנים למעלות, הכפל את הערך המוחזר ע"י ASN ב־18Ø/PI.

דוגמה

הפקודה

```
PRINT 18Ø/PI * ASN Ø.5
```

מציגה 3Ø, ערך הזוית במעלות אשר הסינוס שלה הוא Ø.5.

פורמט

ASN num-const
ASN num-var
ASN (num-expr)

AT

מיקום במקלדת

SYMBOL SHIFT I

ראה: PRINT; LPRINT; INPUT

ATN ארק טנגנס

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT E

פונקציה

ATN מחשבת את ערך הזווית מתוך הטנגנס שלה.

כיצד להשתמש ב־ATN

ATN מלווה בערך מספרי. בטוי חייב להיות בתוך סוגריים, לדוגמה

```
6Ø LET x = ATN (y * z)
```

הערך המלווה את ATN ‏(y * z במקרה זה) הוא הטנגנס של הזווית הנדרשת. ATN מחזירה את ערך הזווית ברדיאנים. כדי להפוך רדיאנים למעלות, הכפל את הערך המוחזר ע"י ATN ב־18Ø/PI.

דוגמה

הפקודה

```
PRINT 18Ø/PI * ATN 1
```

מציגה 45, הזווית במעלות אשר הטנגנס שלה הוא 1.

פורמט

ATN num-const
ATN num-var
ATN (num-expr)

ATTR מאפיינים

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT

פונקציה

ATTR נותנת את המאפיינים של מיקום תו מוגדר על המסך. אלה הם צבעי הדיו והנייר, מצב בהירות והבהוב של התו במיקום המסויים.

כיצד להשתמש ב־ATTR

ATTR מלווה בשני ערכים מספריים המופרדים בפסיק וסגורים במרכאות, לדוגמה

```
15Ø IF ATTR (v,h) = 115 THEN
  GOSUB 2ØØØ
```

הערך הראשון שאחרי ATTR ‏(v שלעיל) נמצא בתחום שבין Ø עד 23 ומייצג את מספר השורה של המיקום על המסך.

הערך השני (h שלעיל) נמצא בתחום שבין Ø עד 31 ומייצג את מספר העמודה של המיקום. ATTR מחזירה מספר הנע בין Ø ל־255. מספר זה הוא סכום המאפיינים במיקום המוגדר, ומורכב מהגורמים הבאים:

צבע דיו — קוד צבע (Ø עד 7)
צבע נייר — 8 פעמים קוד צבע
בהירות — 64
הבהוב — 128

דוגמה

אם תו נמצא בנקודה 16, 11 ומוצג בצבע דיו 3 (סגול), צבע נייר 6 (צהוב) ובהיר אך לא מהבהב, אזי הפקודה

```
PRINT ATTR (11,16)
```

מציגה 115 (Ø + 64 + 6 × 8 + 3).

ATTR בצורה בינארית

ATTR מחזירה בית אחד שבו סיבית 7 (המשמעותית ביותר) היא 1 עבור הבהוב או Ø במצב רגיל, סיבית 6 היא 1 עבור בהירות או Ø במצב רגיל, סיביות 3 עד 5 הן צבע הנייר (בשיטה בינארית) וסיביות 2 עד Ø הן צבע הדיו.

פורמט

ATTR (num-expr ,num expr)

BEEP

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT Z

הצהרה/פקודה

BEEP גורמת לרמקול להפיק צליל יחיד באורך וגובה נתונים.

כיצד להשתמש ב־BEEP

ניתן להשתמש ב־BEEP ליצירת הצהרה בתכנית או פקודה ישירה. BEEP מלווה בשני ערכים מספריים, המופרדים בפסיק, לדוגמה

```
8Ø BEEP x,y
```

הערך הראשון (x) נמצא בתחום Ø עד 1Ø ומגדיר את משך זמן הצליל בשניות.

הערך השני (y) נמצא בתחום 6Ø- עד 69 ומגדיר את גובה הצליל בחצאי טונים. מתחת לדו אמצעי אם הוא שלילי או מעל דו אמצעי אם הוא חיובי.

דוגמה

הפקודה

```
BEEP Ø.5,1
```

גורמת לצליל דו דיאז שמעל דו אמצעי להישמע במשך חצי שנייה.

פורמט

BEEP num-expr, num-expr

BIN מספר בינארי

מיקום במקלדת

EXTEND MODE
B

BIN הופכת מספר בינארי למספר עשרוני.

כיצד להשתמש ב־BIN

BIN מלווה במספר בינארי המורכב מעד 16 תווי 1 ותווי Ø, לדוגמה

```
5Ø POKE USR "a" , BIN 1Ø1Ø
  1Ø1Ø
```

BIN מחזירה את הערך העשרוני של המספר הבינארי. בד"כ משתמשים בו בשילוב עם POKE ו־USR כלעיל ליצירת תוים גרפיים בהגדרת משתמש, כאשר 1 מייצג פיקסל של צבע דיו ו־Ø פיקסל של צבע הנייר.

דוגמה

הפקודה

```
PRINT BIN 1111111Ø
```

מציגה 254, הערך העשרוני של המספר הבינארי.

פורמט

BIN [1][Ø]

BORDER

מיקום במקלדת

B

הצהרה/פקודה

BORDER מגדירה את צבע גבולות המסך סביב שטח התצוגה.

כיצד להשתמש ב־BORDER

BORDER ניתנת לשימוש כפקודה ישירה או כהצהרה בתכנית. BORDER מלווה בערך מספרי, לדוגמה

```
3Ø BORDER RND * 7
```

הערך המלווה את BORDER מעוגל לשלם הקרוב ביותר ומגדיר את צבע גבולות המסך כלהלן:

Ø שחור
1 כחול
2 אדום
3 סגול
4 ירוק
5 תכלת
6 צהוב
7 לבן

שים לב לכך ש־BORDER קובעת גם את צבע הנייר בתחתית המסך. שלא כמו INK ו־PAPER, לא ניתן להכניס הצהרת BORDER בתוך הצהרת PRINT.

פורמט

BORDER int-num-expr

BRIGHT

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT B

הצהרה/פקודה

BRIGHT גורמת לתוים להיות מוצגים בצבעים בהירים יותר מהרגיל.

כיצד להשתמש ב־BRIGHT

BRIGHT ניתנת לשימוש כפקודה ישירה אך משמשת בד"כ ליצירת הצהרה בתכנית. BRIGHT מלווה בערך מספרי, לדוגמה

8Ø BRIGHT 1

הערך המלווה את BRIGHT מעוגל לשלם הקרוב ביותר במידת הצורך, ולכן יהיה Ø, 1 או 8. ערך של 1 גורם לכל התוים הבאים אחריו ומוצגים ע"י PRINT או INPUT להופיע בצבעי דיו ונייר בהירים יותר, וערך של 8 גורם למיקומי תוים בהירים להישאר בהירים ולמיקומי תוים רגילים להישאר רגילים כאשר יודפסו עליהם תוים חדשים.

BRIGHT מלווה ב־Ø מבטלת את BRIGHT 1 ו־BRIGHT 8 כך שכל התווים שיוצגו אחריהם יהיו רגילים.

ניתן להטמיע (להכניס) את BRIGHT בתוך הצהרות תצוגה הנוצרות ע"י PRINT, INPUT, PLOT, DRAW ו־CIRCLE. BRIGHT מלווה את מילת המפתח אך מקדימה את הנתונים או מאפייני התצוגה; היא מלווה באותם ערכים ובנקודה־פסיק; לדוגמה

5Ø PRINT BRIGHT 1;
"WARNING"

ההשפעה של BRIGHT היא איפוא מקומית וישימה רק לתוים המוצגים, נקודות שיותוו, או קווים שיצוירו ע"י הצהרת התצוגה. שים לב לכך ש־BRIGHT 1 מבהירה את צבע הנייר של כל מיקום התו של 8×8 פיקסלים אם אחד הפיקסלים בנקודה זו מותווה בצבע דיו.

פורמט

BRIGHT int-num-expr [;]

CAT קטלוג

פקודת טיפול בקבצים עבור כונן זעיר. ראה מדריך כונן זעיר ומימשק 1.

CHR $ מחרוזת תוים

מיקום במקלדת

EXTEND MODE
U

פונקציה

התוים ומילות המפתח הקיימים במקלדת יחד עם תוים גרפיים בהגדרת המשתמש יוצרים את מערך התוים בספקטרום.

ע"י שימוש ב־CHR $ ומספר קוד, ניתן לקבל כל אחד כמחרוזת. מערך התוים כולל גם כמה קודי בקרה אשר משפיעים על אופן תצוגת התוים. קודים אלה ניתנים להפעלה לקבלת תוים מוצגים בעזרת PRINT לפני CHR $. ניתן למצוא את מערך התוים השלם עם קודי הבקרה עמ' 51.

כיצד להשתמש ב־CHR $

CHR $ מלווה בערך מספרי, לדוגמה

8Ø PRINT CHR$ x

בטוי חייב להיות בתוך סוגריים. הערך המלווה את CHR$ (x לעיל) מעוגל לשלם הקרוב ביותר. אם הוא בתחום של 32 עד 255, CHR$ מחזירה תו של מילת מפתח, תו גרפי בהגדרת משתמש, או מילת מפתח כמחרוזת. הספקטרום משתמש בקוד ASCII עבור ערכים בתחום 32 עד 95 ו־97 עד 126. לדוגמה, אם ל־x מוקצה ערך של 65, ההצהרה שלעיל תציג A.

קודי בקרה של CHR$

ערכים שבין 1 ו־31 מחזירים קודי בקרה או שאינם בשימוש. CHR$ 6 (PRINT פסיק), 8 (מקש־אחורה) ו־13 (שורה חדשה או ENTER) משפיעים על תצוגות במסך אם יוכללו בהצהרת PRINT. ניתן ללוות CHR$ בקוד הבקרה ונקודה־פסיק, לדוגמה

6Ø PRINT "A" ; CHR$ 6 ; "B"

הצהרה זו מציגה

A B

דרך אחרת להשתמש בקודי בקרה של CHR$ הוא ליצור מחרוזת מורכבת המכילה אותם. ההצהרה

6Ø PRINT "A"+CHR$6+"B"

נותנת בדיוק אותו אפקט כמו הדוגמה הקודמת.

קודים 16 עד 23 משפיעים על צבע ומיקום וניתן להשתמש בכל אחד מהם במחרוזת מורכבת יחד עם CHR$ המלווה בערך קוד צבע בתחום Ø עד 7 עבור 16 CHR$ (בקרת דיו) ו־17 CHR$ (בקרת נייר), או ע"י Ø או 1 עבור 18 CHR$ עד 21 CHR$ (בקרי FLASH, BRIGHT, INVERSE ו־OVER). הפקודה

PRINT CHR$ 16 + CHR$ 3 + CHR$ 17 + CHR$ 6 + CHR$ 18 + CHR$ 1 + "ZX SPECTRUM + "

מציגה ZX SPECTRUM + בצבעי אדום וצהוב מהבהבים.

לחלופין, כמו לעיל, כל סימן פלוס (+) ניתן להחלפה בנקודה־פסיק.

22 CHR$ (בקר AT) מלווה בשני ערכי CHR$ לציון מספרי השורה ועמודה. הפקודה

PRINT CHR$ 22 + CHR$ 11 + CHR$ 16 + CHR$ 42

מציגה כוכב במרכז המסך.

23 CHR$ (בקרת TAB) מלווה גם היא בשני ערכים באותה צורה. הערך השני הוא בד"כ Ø והראשון נותן את מיקום TAB. הפקודה

PRINT CHR$ 23 + CHR$ 16 + CHR$ Ø + CHR$ 42

מציגה כוכב במחצית הדרך לרוחב המסך.

שים לב לכך שקיימים רק בקרים אלה.

שימוש ב־PRINT CHR$ עם ערך של מילת מפתח גדול מ־164 מציג רק את מילת המפתח אך לא מפעיל אותה.

פורמט

CHR$ int-num-const [;] [+]
CHR$ int-num-var [;] [+]
CHR$ int-num-expr([;] [+]

CIRCLE

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT H

הצהרה/פקודה

CIRCLE מציירת עיגול על המסך.

כיצד להשתמש ב־CIRCLE

CIRCLE מלווה ב־3 ערכים מספריים המופרדים בפסיקים, לדוגמה

8Ø CIRCLE x,y,z

כל אחד משלשת הערכים מעוגל לשלם הקרוב ביותר במידת הצורך.

CIRCLE מצייר עיגול על רשת הנקודות בהפרדה גבוהה בצבע הדיו הקיים.

שני הערכים הראשונים (x,y) מגדירים את הקואורדינטות האפקית והאנכית של המרכז, והערך השלישי (z) מגדיר את אורך הרדיוס. המימדים חייבים להיות כאלה שלא יחרגו משטח התצוגה.

CIRCLE מושפעת מהצהרות צבע או פקודות ועשוייה לכלול הצהרות צבע מוטמעות עם אותן תוצאות כמו PLOT ו־DRAW.

דוגמה

הפקודה

```
CIRCLE 128,88,87
```

מציירת מעגל התופס את מרבית שטח התצוגה.

פורמט

CIRCLE [statement ;] int-num-expr , int-num-expr , int-num-expr

CLEAR

מיקום במקלדת

X

הצהרה/פקודה

CLEAR מוחקת את הערכים הקיימים של כל המשתנים, משחררת את מרחב הזכרון שהערכים תפסו ומרחב עד מחיצת RAMTOP בזכרון, שהיא הכתובת העליונה במרחב מערכת בייסיק.

ניתן גם להשתמש ב־CLEAR כדי לאפס את RAMTOP.

כיצד להשתמש ב־CLEAR

ניתן להשתמש ב־CLEAR כפקודה ישירה או בתור הצהרה בתכנית. לא נדרשים כל ערכים, לדוגמה

```
50 CLEAR
```

במקרה זה CLEAR תמחק את כל הערכים המוקצים לכל המשתנים, כולל מערכים. כמו כן היא תבצע CLS ו־RESTORE במטרה לנקות את המסך ולהחזיר את סמן הנתונים לפריט המידע הראשון. בנוסף, מיקום PLOT מוחזר לפינה שמאלית תחתונה של שטח התצוגה ומחסנית GOSUB מתרוקנת.

שים לב שלא נדרשת פקודת CLEAR לפני הגדרת מימדי מערכים מחדש מפני ש־DIM מוחקת מערך קיים באותו שם. שים לב גם לכך ש־RUN מבצעת CLEAR.

CLEAR ו־RAMTOP

ניתן ללוות את CLEAR בערך מספרי, לדוגמה

```
CLEAR 65267
```

במקרה זה CLEAR מבצעת מה שתואר לעיל וגם מביאה את RAMTOP, הכתובת הגבוהה ביותר של מרחב מערכת בייסיק, לערך הנתון. RAMTOP נקבעת ב־65367 בספקטרום +ZX, ונמצאת מתחת לאיזור השמור עבור גרפיקה בהגדרת־משתמש.

NEW מנקה את הזכרון עד ל־RAMTOP ולכן, השימוש ב־CLEAR להנמכת מיקום RAMTOP (ב־100 בתים בדוגמה שלעיל) מאפשר יותר זכרון החסין בפני NEW.

העלאת RAMTOP נותנת מרחב גדול יותר לבייסיק על חשבון גרפיקה בהגדרת־משתמש.

שים לב לכך שמחסנית GOSUB תמוקם אזי ב־RAMTOP, בתנאי ש־RAMTOP נשארת מעל מחסנית המחשבון.

כדי למצוא את RAMTOP, הקש

```
PRINT PEEK 23730 + 256 *
  PEEK 23731
```

פורמט

CLEAR [num-expr]

CLOSE

פקודת טיפול בקבצים בכונן זעיר. ראה מדריך שימוש בכונן זעיר ומימשק 1.

CLS נקה מסך

מיקום במקלדת

V

פקודה/הצהרה

CLS מוחקת את כל הטקסט והגרפיקה משטח התצוגה, ומותירה אותו נקי בצבע הנייר (רקע) הקיים.

כיצד להשתמש ב־CLS

ניתן להשתמש ב־CLS כפקודה ישירה או בתור הצהרה בתכנית. לא נדרשים כל נתונים נוספים, לדוגמה

```
250 IF a$ = "NO" THEN CLS
```

שטח התצוגה (אך לא הגבולות) מתנקה ועובר לצבע שנבחר ע"י הצהרת PAPER או פקודה הקודמת או לצבע הנייר בברירת המחדל, הלבן.

שים לב שחייבים להשתמש ב־CLS לאחר PAPER ולפני PRINT או כל הצהרת תצוגה אחרת ליצירת רקע צבעוני על פני כל שטח התצוגה.

פורמט

CLS

CODE

מיקום במקלדת

EXTEND MODE

I

פונקציה

CODE נותנת את מספר הקוד של תו במערך התוים של הספקטרום (ראה עמ' 51).

כיצד להשתמש ב־CODE

CODE מלווה בערך מחרוזת, לדוגמה

```
90 IF CODE a$ < 65 OR CODE
  a$ < 90 THEN GOTO 80
```

בטוי מחרוזת חייב להיות בתוך סוגריים. CODE מחזירה את מספר הקוד של התו הראשון במחרוזת. אם זוהי מחרוזת ריקה (""), אזי CODE מחזירה 0.

SAVE CODE ו־LOAD CODE

השימוש ב־CODE עם SAVE ו־LOAD הוא בצורה שונה. ראה הסברים על SAVE CODE ו־LOAD CODE.

פורמט

CODE string-const
CODE string-var
CODE (string-expr)

CONTINUE

מיקום במקלדת

C

פקודה

אם תכנית נעצרת, ניתן להשתמש ב־CONTINUE להתחיל מחדש את התכנית מהנקודה בה היא נעצרה. אם אירעה שגיאה שגרמה להפסקת התכנית, אזי חייבים לתקנה לפני ש־CONTINUE תאפשר לתכנית להמשיך.

כיצד להשתמש ב־CONTINUE

CONTINUE משמשת כפקודה ישירה כאשר תכנית נעצרה. אין צורך בנתונים נוספים. לאחר CONTINUE התכנית ממשיכה באופן רגיל באותה הצהרה בה היא נעצרה. אם הסיבה היתה שגיאה, אזי ניתן להקיש פקודה לתיקון השגיאה ו־CONTINUE תאפשר לתכנית להמשיך מההצהרה. אם התכנית נעצרה בהצהרת STOP במתן דוח 9 או אם נעצרה עקב הקשת BREAK וקבלת דוח L, אזי CONTINUE גורמת לתכנית להמשיך מההצהרה הבאה. ניתן להכניס תחילה פקודה מתקנת, במידת הצורך. אם

משתמשים ב־CONTINUE להמשך פקודה ישירה, אזי היא תכנס ללולאה אם הפקודה נעצרה בהצהרה הראשונה שבה. התצוגה נעלמת, אך ניתן להחזיר את הבקרה בלחיצת BREAK. CONTINUE נותנת דווח 0 אם הפקודה נעצרה בהצהרה השנייה ודווח N בהצהרה השלישית או בהצהרות הבאות.

פורמט

CONTINUE

COPY

מיקום במקלדת

Z

פקודה

COPY גורמת למדפסות מסוג סינקלייר להפיק עותק מודפס של תצוגת המסך.

כיצד להשתמש ב־COPY

COPY משמשת בתור פקודה ישירה כאשר תכנית הסתיימה או נעצרה. היא לא דורשת כל נתונים. לאחר COPY, ובתנאי שהמדפסת מחוברת, יודפס עותק של 22 השורות הראשונות במסך. שים לב לכך שכל צבעי הדיו מודפסים בשחור; צבעי הנייר (רקע) אינם מודפסים. ניתן לעצור את המדפסת בלחיצת BREAK. אם מופיע רישום תכנית על המסך, ניתן להדפיסו ע"י COPY בתנאי שנוצר בפקודת או הצהרת LIST. שים לב לכך שהקשת ENTER לאחר שתכנית הסתיימה או נעצרה תגרום להופעת רישום על המסך, אך לא ניתן להדפיס רישום "אוטומטי" זה באמצעות COPY.

פורמט

COPY

COS קוסינוס

מיקום במקלדת

EXTEND MODE
W

פונקציה

COS נותנת את קוסינוס הזוית.

כיצד להשתמש ב־COS

COS מלווה בערך מספרי, לדוגמה

```
140 LET x = COS y
```

בטוי חייב להיות בתוך סוגריים. הערך המלווה את COS הוא הזוית ברדיאנים. COS מחזירה את קוסינוס הזוית. ניתן להפוך מעלות לרדיאנים ע"י הכפלה ב־PI/180.

שים לב ש־COS מחזירה ערך שלילי עבור זויות בתחום 90 עד 270 מעלות וערך חיובי עבור זויות בתחום 0 עד 90 ו־270 עד 360 מעלות.

דוגמה

הפקודה

```
PRINT COS (60 * PI/180)
```

מציגה 0.5, קוסינוס זוית בת 60 מעלות.

פורמט

COS num-const
COS num-var
COS (num-expr)

DATA

מיקום במקלדת

EXTEND MODE
D

הצהרה

DATA מספקת רשימת פריטי מידע בתוך תכנית. פריטים אלה יכולים להיות, לדוגמה, ערכי משתנים או מחרוזות שיש להציג. כל פריט מוקצה לטובת משתנה ע"י הצהרת READ.

ההקצאה מתבצעת בסדר בו מופיעים פריטי המידע בתכנית, אך ניתן להשתמש ב־RESTORE להתחלת הקצאה בפריט הראשון שבהצהרת DATA נתונה.

כיצד להשתמש ב־DATA

ניתן להשתמש ב־DATA ליצירת הצהרה בתכנית בלבד. בד"כ היא מלווה ברשימת קבועים מספריים או מחרוזות, המופרדים כולם בפסיקים, לדוגמה

```
50 DATA 31 , "JAN" , 28 , "FEB"
```

כל קבוע מוקצה במקרה זה למשתנה ע"י הצהרת READ הקוראת את DATA. הצהרת DATA ניתנת למיקום בכל מקום בתכנית. המספר, סוג (מספרי או מחרוזת) וסדר הקבועים חייב להתאים למספר הפעמים בו מתבצעת הצהרת READ וסוג וסדר המשתנים בהצהרת READ. ניתן לפצל את רשימת המידע לכמה הצהרות DATA רצופות במקרה של פריטים רבים מדי בהצהרה אחת.

דוגמה

התכנית הבאה:

```
10 FOR n = 1 TO 2
20 READ x , a$
30 PRINT a$ , x ; "days"
40 NEXT n
50 DATA 31 , "JAN" , 28 , "FEB"
```

מציגה

JAN	31 days
FEB	28 days

שימוש ב־DATA עם משתנים

פריטי המידע בהצהרת DATA עשויים לכלול משתני מחרוזת או מספרים, או ביטויים בתנאי שלמשתנים הוקצו מקודם ערכים. בדוגמה שלעיל, ניתן לשנות את הצהרת DATA בצורה הבאה:

```
50 DATAd,m$,d-3, "FEB"
```

אם הוקצה מקודם ל־d ערך של 31 ול־m$ המחרוזת "JAN", תושג אותה התצוגה על המסך.

LOAD DATA, SAVE DATA ו־VERIFY DATA

DATA ניתנת לשימוש גם עם LOAD, SAVE, ו־VERIFY לצורך איחסון מערכים על סרט. ראה LOAD DATA, SAVE DATA ו־VERIFY.

פורמט

DATA num-expr [, num-expr]
[, string-expr]
DATA string-expr [, num-expr]
[, string-expr]

DEF FN הגדר פונקציה

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT 1

הצהרה

DEF FN מאפשרת למשתמש להגדיר פונקציה שאינה קיימת כמילת מפתח. ניתן להעביר מבחר גדול של נתונים לפונקציה בהצהרת FN, אשר קוראת לפונקציה ועשוייה להחזיר ערך מספרי או מחרוזת בתור תוצאה.

כיצד להשתמש ב־DEF FN

ניתן להשתמש ב־DEF FN כהצהרה בתכנית בלבד. אם יש להגדיר פונקציה מספרית, DEF FN מלווה במילה בודדת כלשהי ואח"כ במשתנה מספרי אחד או יותר, כשכל אחד

מופרד בפסיק ונמצאים בתוך סוגריים, לדוגמה DEF FN r(x,y) כל זה מלווה בסימן שוויון ואחריו בטוי מספרי המכיל את המשתנים, לדוגמה

```
1000 DEF FN r(x , y) =
  SQR (x↑2 + y↑2)
```

האות המלווה את DEF FN (r שלעיל) היא שם זיהוי לפונקציה. גם המשתנים יכולים להיות רק אותיות בודדות. שים לב לכך שבשני המקרים, הספקטרום אינו מבחין בין אותיות גדולות וקטנות. הביטוי שאחרי סימן השוויון משתמש במשתנים (x ו־y שלעיל) להגדרת הפונקציה.

ניתן למקם הצהרת DEF FN בכל מקום בתכנית. כדי לקרוא לפונקציה המוגדרת, משתמשים בהצהרת FN. ההצהרה תלווה באות שם הפונקציה ורשימת ערכים מספריים בתוך סוגריים ומופרדים בפסיקים, לדוגמה

```
50 PRINT FN r(3 , 4)
```

הערכים שבתוך הסוגריים מועברים לפונקציה באותו סדר כמו המשתנים בהצהרת DEF FN. לכן, בדוגמה זאת, ל־x מוקצה ערך של 3 ול־y ערך של 4. FN מחשבת את הביטוי ומחזירה את הערך.

ניתן גם ללוות את DEF FN באות וצמד סוגרים בלבד, לדוגמה

```
1000 DEF FN r( ) = INT(x + 0.5)
```

הערך המוקצה כעת למשתנה (x לעיל) מועבר לפונקציה כאשר היא נקראת ע"י FN. במקרה זה, FN r( ) מחזירה את הערך המוקצה כעת ל־x כשהוא מעוגל לשלם הקרוב ביותר.

DEF FN ומחרוזות

ניתן להשתמש ב־DEF FN ו־FN באותה דרך גם כדי להגדיר ולקרוא לפונקציית מחרוזת. במקרה זה, שם הפונקציה הוא אות בודדת בלווית $ ומשתנה אחד או יותר בפונקציה הוא אות בלווית $.

ביטויי מחרוזת מתאימים יוצרים את ההגדרה, לדוגמה

```
1000 DEF FN a$(b$ ,x , y) =
  b$ (x TO y)
```

ביטוי המחרוזת הבא לאחר סימן השוויון בדוגמה זו הוא פורס מחרוזת, ו־x ו־y הם התו הראשון והתו האחרון של חלק מ־b$. FN חייב להיות מלווה בשם הפונקציה, וערך מחרוזת יחד עם כל נתון שיש להעבירו לפונקציה בתוך סוגריים. במקרה זה, הפקודה

```
PRINT FN a$
("FUNDAMENTAL",1,3)
```

מציגה FUN, ואילו הפקודה

```
PRINT FN a$
("FUNDAMENTAL",5,8)
```

מציגה AMEN

פורמט

```
DEF FN letter ([letter][, letter])
= num-expr
DEF FN letter$ ([letter$][letter]
[, letter][, letter$]) = string-expr
FN letter ([num-expr][, num-expr])
FN letter$ ([string-expr][num-expr]
[,num-expr][,string-expr])
```

DIM מימד

מיקום במקלדת

D

הצהרה

DIM משמשת למתן מימדים (הכנה) למערך בעל מספר נתון של משתני מספרים או מחרוזות.

מערך הוא רשימת משתנים באותו שם אשר מזוהים ע"י תגים (ערכים אשר מזהים כל משתנה או איבר במערך).

כיצד להשתמש ב־DIM עם מערכים מספריים

DIM משמשת ליצירת הצהרה בתכנית. היא מלווה באות בודדת הנותנת שם למערך, ואחד או יותר ערכים מספריים המופרדים בפסיקים, ובתוך סוגריים, לדוגמה

```
20 DIM x (10)
80 DIM z (20 , 5)
```

במקרה הראשון, נוצר מערך מספרי חד־מימדי המכיל 10 איברים עם תגים מ־1 עד 10. שם המערך הוא x והמשתנים המותגים הם x(1) עד x(10) כולל. כל מערך הקיים באותו שם נמחק, ולמשתנים מוקצה ערך 0. שים לב לכך שבמתן מימדים למערך, הספקטרום אינו מבחין בין שמות עם אותיות גדולות וקטנות — משתנה x(2) זהה ל־X(2). ברם, משתנים מספריים פשוטים בעלי אותה אות כמו שם מערך (x או X) יכולים להתקיים יחד וניתנים לשימוש בנפרד, במידת הצורך.

מספר הערכים בסוגריים שווה למספר המימדים הנוצרים במערך מספרי. הדוגמה השנייה מכינה מערך דו־מימדי בן 100 איברים עם 20 איברים במימד הראשון ו־5 בשני. איברים אלה ממוספרים z(1,1) עד z(20,5). ניתן ליצור מערכים בכל מספר מימדים.

איברי מערך מספרי ניתנים לזיהוי **לפי הסדר ע"י שם המערך בלווית ערך בסוגריים, לדוגמה**

```
70 PRINT x(a)
160 PRINT z(7 , b)
```

DIM ומערכי מחרוזות

ניתן להשתמש ב־DIM באותה צורה כמו עם מערכים מספריים פרט לכך שעבור שם המערך משתמשים באות בודדת המלווה בסימן $.

כמו כן, יש להוסיף ערך נוסף לערכי המימדים בסוגריים במטרה להגדיר אורך כל מחרוזת, לדוגמה

```
30 DIM a$(20 , 5)
90 DIM b$(20 , 5 , 10)
```

ההצהרה הראשונה יוצרת מערך בן 20 איברים, כשכל אחד מכיל מחרוזת בת 5 תוים. המשתנים המותגים נקראים a$(1) עד a$(20) כולל, ובתחילה מוקצה להם מחרוזת אפס (ריקה) (""). כל מערך קיים באותו שם נמחק, ולא כמו במערכים מספריים, לא יכול להתקיים משתנה מחרוזת פשוט נוסף באותו שם.

הדוגמה השנייה יוצרת מערך מחרוזות דו־מימדי בן 100 איברים עם 20 איברים במימד הראשון ו־5 במימד השני. כל האיברים באורך של 10 תוים. כאשר ערכי מחרוזת מוקצים לפי הסדר למערך מחרוזות, הם "מרופדים" במרווחים בסוף המחרוזת או מתקצצים לאורך המוגדר, במידת הצורך.

איברי מערך מחרוזות מזוהים ע"י שם המערך המלווה, בסוגריים, באחד או יותר ערכים מספריים הנותנים את מספר(י) התג. לדוגמה, איבר a$(2) עשוי להיות "SMITH" ואיבר b$(12,4) עשוי להיות "DERBYSHIRE". ברם, ניתן להוסיף ערך נוסף להגדרת תו מסויים במחרוזת. בדוגמאות אלה, a$(2,2) יהיה "M" (התו השני ב־"SMITH") ו־b$(12,4,5) יהיה "Y".

ערכי מחרוזות במימד־אפס
ניתן ליצור מערך מחרוזות במימד־אפס ע"י שימוש בערך אחד בלבד בסוגריים, לדוגמה

```
1Ø DIM c$(15)
```

למערך זה יש איבר אחד בלבד, שהוא c$, וארכו קבוע בערך המוגדר (15 תוים).

פורמט
DIM letter (num-expr [, num-expr])
DIM letter$ (num-expr [, num-expr])

DRAW

מיקום במקלדת
W

הצהרה/פקודה
DRAW משמשת לציור קוים ישרים ועקומות על המסך.

כיצד להשתמש ב־DRAW
DRAW משמשת בד"כ ליצירת הצהרה בתכנית. אם נדרש קו ישר, היא מלווה בשני ערכים מספריים המופרדים בפסיק, לדוגמה

```
4Ø DRAW x, y
```

קו ישר יצוייר על רשת נקודות הגרפיקה בהפרדה גבוהה מהנקודה המוגדרת ע"י הצהרת PLOT הקודמת או הנקודה אליה הגיעה הצהרת DRAW הקודמת, מה שבא אחרון.

שני הערכים המלווים את DRAW מעוגלים לשלם הקרוב ביותר, אם נדרש.

הערך הראשון (x לעיל) מגדיר את המרחק האפקי מנקודה זאת, והערך השני (y) את המרחק האנכי. ערכים אלה הם שליליים אם הקו פונה שמאלה או מטה בהתאמה, ונקודת היעד חייבת להימצא בגבולות שטח התצוגה.

אם לא קיימת הצהרת PLOT או DRAW קודמת, DRAW מתחילה מנקודה Ø,Ø (פינה שמאלית תחתונה במסך).

DRAW מושפעת מהצהרות או פקודות צבע ועשוייה לכלול הצהרות מוטמעות עם אותם האפקטים כמו עם PLOT ו־CIRCLE.

ציור קוים עקומים
ניתן להוסיף ל־DRAW ערך שלישי ליצירת עקומה המהווה חלק מעיגול, לדוגמה

```
4Ø DRAW x , y , z
```

הערך השלישי (z שלעיל) מגדיר את הזוית (ברדיאנים) שאותה עובר הקו תוך כדי ציורו. הקו מסתובב שמאלה אם הערך חיובי, וימינה אם הערך שלילי. ערכים של PI או -PI יוצרים עיגול.

דוגמה
התכנית הבאה מציירת משולש:

```
1Ø PLOT 127 , 15Ø
2Ø DRAW 7Ø , -1ØØ
3Ø DRAW -14Ø , Ø
4Ø DRAW 7Ø , 1ØØ
```

הוספת 1 או -1 ליצירת DRAW גורמת לצדדים להתעקם פנימה או החוצה, בהתאמה.

פורמט
DRAW [statement ;] int-num-expr
int-num-expr [, int-num-expr]

ERASE

פקודת טיפול בקבצים בכונן זעיר. ראה מדריך כונן זעיר ומימשק 1.

EXP אקספוננט

מיקום במקלדת
EXTEND MODE
X

פונקציה
EXP היא פונקציה מתימטית המעלה את האקספוננט e בחזקה הנתונה.

כיצד להשתמש ב־EXP
EXP מלווה בערך מספרי, לדוגמה

```
6Ø LET y = EXP x
```

ביטוי חייב להיות בתוך סוגריים. EXP מחזירה את האקספוננט e כשהוא מועלה בחזקת הארגומנט (x לעיל).

דוגמה
הפקודה

```
PRINT EXP 1
```

מציגה 2.7182818, הערך של e.

פורמט
EXP num-const
EXP num-var
EXP (num-expr)

FLASH

מיקום במקלדת
EXTEND MODE
SYMBOL SHIFT V

הצהרה/פקודה
FLASH גורמת להבהוב במיקומי התוים, בכך שצבעי הדיו והנייר מתחלפים בקצב קבוע.

כיצד להשתמש ב־FLASH
ניתן להשתמש ב־FLASH כפקודה ישירה אך בד"כ היא יוצרת הצהרה בתכנית. היא מלווה בערך מספרי, לדוגמה

```
5Ø FLASH 1
```

הערך הבא אחרי FLASH מעוגל לשלם הקרוב ביותר במידת הצורך ועשוי להיות Ø, 1 או 8. ערך של 1 גורם לכל התוים שיוצגו בהמשך ע"י PRINT או INPUT להבהב. ערך של 8 גורם למיקומי התוים המהבהבים להמשיך ולהבהב, ולמיקומי תוים רגילים להישאר במצבם גם לאחר הכנסת תוים חדשים. FLASH בלווית Ø מבטלת את FLASH 1 וגם את FLASH 8 כך שכל התוים שיוצגו בהמשך הם רגילים. ניתן גם להטמיע (להכניס) FLASH בתוך הצהרות תצוגה הנוצרות ע"י PRINT, INPUT, PLOT, DRAW ו־CIRCLE. FLASH מלווה את מילת המפתח אך מקדים את נתוני המידע או התצוגה; הוא מלווה באותם ערכים ונקודה־פסיק, לדוגמה

```
12Ø PRINT FLASH 1 ; INK 2 ;
  PAPER 6 ; "WARNING"
```

ל־FLASH יש, איפוא, השפעה מקומית הישימה רק לתוים המוצגים, נקודות מותוות או קוים שצוירו בהצהרת תצוגה. שים לב לכך ש־FLASH 1 גורמת לכל 8×8 מיקומי הפיקסלים להבהב אם אחד מהפיקסלים מותווה בצבע דיו.

פורמט

FLASH int-num-expr[;]

FN פונקציה

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT 2

פונקציה

FN קוראת לפונקציה בהגדרת־משתמש. היא בשימוש תמיד בקשר עם DEF FN, אשר מגדירה את הפונקציה הנקראת.

כיצד להשתמש ב־FN

אם יש לקרוא לפונקציה מספרית, FN מלווה באות ואחריה זוג סוגריים. אם יש להעביר נתונים מסויימים לפונקציה, הם יוכנסו, מופרדים בפסיקים, בתוך הסוגריים, לדוגמה

```
17Ø LET x = FN r(3 , 4)
```

הנתונים (3 ו־4 שלעיל) מועברים לפונקציה הנקראת r. FN מחזירה את התוצאה. אם אין נתונים להעברה, עדיין יש לכלול את צמד הסוגריים, לדוגמה

```
7Ø PRINT FN r( )
```

במקרה זה, הפונקציה משתמשת בערכים המוקצים באותו זמן למשתניה. FN קוראת לפונקציית מחרוזת באותה צורה, פרט לכך שיש להוסיף סימן $ אל האות של שם הפונקציה. ראה פרטים נוספים ב־DEF FN.

פורמט

FN letter ([num-expr] [, num-expr])
FN letter$ ([string-expr] [num-expr] [, num-expr] [, string-expr])

FOR

מיקום במקלדת

F

הצהרה/פקודה

FOR פועלת תמיד יחד עם מילות המפתח TO ו־NEXT ליצירת לולאת FOR NEXT. מבנה זה מאפשר חזרה על חלק מסויים בתכנית מספר נתון של פעמים.

כיצד להשתמש ב־FOR

FOR משמשת תמיד ליצירת הצהרה יחד עם TO. FOR מלווה באות, סימן שוויון, ואחריהם שני ערכים מספריים המופרדים ע"י TO, לדוגמה:

```
6Ø FOR a = 1 TO 9
```

האות (a שלעיל) יוצרת משתנה בקרה. אח"כ באות ההצהרות שיש לחזור על ביצוען, כאשר אחת או יותר מהן מנצלת בד"כ את משתנה הבקרה. הלולאה מסתיימת בהצהרת NEXT, שבה NEXT מלווה במשתנה הבקרה. לדוגמה

```
9Ø NEXT a
```

בעת הביצוע, FOR מוחק את כל המשתנים בעלי אותו שם כמו משתנה הבקרה ומקצה לו ערך ראשוני שיבוא לפני TO (1 לעיל). ההצהרות מתבצעות כאשר משתנה הבקרה מכיל ערך זה. כאשר מגיעים ל־NEXT, ערך משתנה הבקרה עולה ב־1. אם ערך זה קטן או שווה לערך שאחרי TO (ערך הגבול של 9 לעיל), התכנית חוזרת להצהרת FOR ולולאת FOR NEXT חוזרת. אם משתנה הבקרה מקבל ערך גדול מערך הגבול, הלולאה מסתיימת והתכנית ממשיכה בהצהרה הבאה אחרי NEXT.

בדוגמה שלעיל, הלולאה חוזרה 9 פעמים כשמשתנה הבקרה משתנה מ־1 עד 9. ביציאה מהלולאה, a הוא בערך של 1Ø.

שים לב לכך שהספקטרום אינו מבחין בין אותיות גדולות וקטנות בעת קריאת שם למשתנה הבקרה.

שימוש ב־STEP בלולאת FOR NEXT

STEP היא מילת מפתח אשר ניתן לשלבה בהצהרת FOR אם יש להגדיל את משתנה הבקרה בערך שונה מ־1 או להפחיתו.

STEP מלווה את ערך הגבול ומלווה בערך מספרי, לדוגמה

```
6Ø FOR a = 1 TO 9 STEP 2
```

משתנה הבקרה מוגדל בערך המדרגה (2 לעיל) עד שהוא גדול מערך הגבול. משתנה הבקרה מקבל ערכים עוקבים של 1, 3, 5, 7 ו־9 ועוזב את הלולאה בערך של 11.

ערך מדרגה שלילי גורם להפחתת ערך משתנה הבקרה. במקרה זה, הערך הראשוני חייב להיות גדול מזה של ערך הגבול והלולאה מסתיימת כאשר ערך משתנה הבקרה נמוך מזה של ערך הגבול, לדוגמה

```
6Ø FOR a = 9 TO 1 STEP -1
```

ערך a יורד מ־9 עד 1, ו־a עוזב את הלולאה כאשר ערכו Ø.

לולאות מקננות

ניתן למקם אחת או יותר לולאות FOR NEXT אחת בתוך השנייה, תהליך הנקרא "קינון" לולאות. סדר משתני הבקרה בהצהרות NEXT חייב להיות בסדר הפוך למשתני הבקרה בהצהרות FOR. לולאות FOR NEXT ניתנות לקינון בכל עומק, כלומר, ניתן למקם לולאות אחת בתוך השנייה ככל שנדרש.

פורמט

FOR letter = num-expr TO num-expr [STEP num-expr] NEXT letter

FORMAT

פקודת טיפול בקבצים בכונן זעיר. ראה מדריך כונן זעיר ומימשק 1.

GOSUB

מיקום במקלדת

H

הצהרה/פקודה

GOSUB גורמת לתכנית להסתעף אל שיגרה, שהיא קטע נפרד בתכנית. הפעולה שימושית אם השיגרה נדרשת מספר פעמים בתכנית.

כיצד להשתמש ב־GOSUB

ניתן להשתמש ב־GOSUB כהצהרה או פקודה ישירה והיא מלווה בערך מספרי, לדוגמה

```
GOSUB 1ØØØ
```

בזמן הביצוע, הערך המלווה את GOSUB (1ØØØ שלעיל) מעוגל לשלם הקרוב ביותר, והתכנית מסתעפת למספר השורה בעלת אותו ערך. השימוש במשתנה או בטוי מאפשר לתכנית להסתעף אל שיגרה במספר שורה מחושב. שים לב לכך שאם מספר השורה לא קיים, התכנית עדיין מסתעפת וממשיכה עם ההצהרה הראשונה שמגיעים אליה. כל שגרה מסתיימת ב־RETURN, והתכנית מסתעפת חזרה אל ההצהרה הבאה אחרי הצהרת GOSUB. ניתן לקנן שגרות כך שמשיגרה אחת מגיעים לשנייה, ובמקרה זה RETURN שולח את התכנית חזרה להצהרה הבאה אחרי הצהרת GOSUB האחרונה שבוצעה.

מחסנית GOSUB

כאשר מתבצעת GOSUB, מספר השורה מועבר למחסנית GOSUB בזכרון. אם מתבצעות שתי הצהרות GOSUB או יותר לפני RETURN, מספרי השורות שלהן נערמים במחסנית כך שהמספר האחרון נמצא בראש המחסנית. RETURN לוקחת תמיד את מספר השורה מראש המחסנית ופונה לשורה זו להמשך התכנית.

שים לב לכך שתיתכן שגיאה 4 (Out of memory) במידה ואין מספיק הצהרות RETURN.

פורמט

GOSUB int-num-expr.

GOTO

מיקום במקלדת

G

הצהרה/פקודה

GOTO גורמת לתכנית להסתעף אל שורה מסוימת.

כיצד להשתמש ב־GOTO

ניתן להשתמש ב־GOTO כפקודה ישירה להרצת תכנית ממספר שורה נתון ללא ניקוי המסך תחילה.

ניתן להשתמש בה גם ליצירת הצהרה בתכנית.

GOTO מלווה בערך מספרי, לדוגמה

```
6Ø GOTO 35Ø
```

בעת הבצוע, הערך המלווה את GOTO מעוגל לשלם הקרוב ביותר והתכנית מסתעפת למספר השורה באותו ערך. השימוש במשתנה או בטוי מאפשר לתכנית להסתעף אל מספר שורה מחושב. שים לב לכך שאם השורה לא קיימת, התכנית עדיין תסתעף ותמשיך בהצהרה הראשונה שתמצא.

פורמט

GOTO int-num-expr

הצהרה/פקודה

IF באה לרוב בצירוף מילת המפתח THEN כדי לפנות להחלטה המשפיעה על הפעולות הבאות. כדי לבצע זאת, המחשב בודק משהו כדי למצוא האם הוא אמת או לא. אם הוא אמת, יופעל נתיב פעולה מסויים. אם הוא לא אמת, יופעל נתיב אחר.

כיצד להשתמש ב־IF ו־THEN

IF יוצר בדרך כלל הצהרה עם THEN. IF מלווה בתחילה בערך מספרי או בתנאי, ואח"כ ב־THEN עם הצהרת בייסיק אחת או יותר, לדוגמה

```
8Ø IF x THEN GOTO 25Ø
24Ø IF a$ = "NO" THEN PRINT
  "THE END" : STOP
```

קבוע, משתנה או בטוי (כמו x לעיל) נחשב אמת אם ערכו שונה מאפס. במקרה זה, מתבצעות ההצהרה הבאה אחרי THEN וכל הצהרה נוספת באותה שורה. התכנית ממשיכה לשורה הבאה. אם הערך הוא Ø, אזי הקבוע, משתנה או בטוי נחשב שקרי. ההצהרות הבאות לא מתבצעות והתכנית מדלגת לשורה הבאה. בדוגמה זו, התכנית לא תפנה לשורה 25Ø אם x שוה Ø.

אם התנאי (a$ = "NO") הבא אחרי IF הוא אמת, אזי מתבצעות ההצהרות הבאות אחרי THEN. אם התנאי שקרי, התכנית תפנה לשורה הבאה. בדוגמה זו, אם a$ הוא בעל ערך "NO" אזי מוצג "THE END" והתכנית נעצרת. אם ל־a$ יש ערך אחר כלשהו, התכנית תמשיך מהשורה הבאה.

הספקטרום נותן לתנאי אמת ערך של 1 ולתנאי שקרי ערך Ø. הוא מכיר כל ערך שונה מאפס כאמת וכל ערך Ø כשקר. ניתן להקצות למשתנה את ערך התנאי בעזרת הצהרה, כמו

```
7Ø LET x = a$ = "NO"
```

שים לב לכך, שלא כמו בניבי בייסיק אחרים, לא ניתן להשמיט את THEN לפני GOTO.

פורמט

IF num-expr THEN statement
[:statement]
IF cond THEN statement
[:statement]

IN

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT I

פונקציה

IN בודקת את מצב המקלדת והתקני קלט ופלט אחרים. היא קוראת בית מכתובת שער נתונה אשר מציינת את מצב ההתקן המחובר לשער.

כיצד להשתמש ב־IN

IN מלווה בערך מספרי, לדוגמה

```
15Ø LET x = IN y
```

הערך המלווה את IN נמצא בתחום Ø עד 65535 ומגדיר את כתובת השער שיש לקרוא. IN מחזירה את הבית הנקרא משער זה.

כתובות מקלדת

למקלדת יש 8 כתובות, שכל אחת מהן עשוייה להכיל אחד מתוך 5 בתים שונים, בהתאם למקש הנלחץ. הכתובות הן 65278, 65Ø22, 6451Ø, 63486, 61438, 57342, 4915Ø ו־32766.

ערכי בית בכתובות אלה עשויים להיות 175, 183, 187, 189 או 19Ø.

פורמט

IN num-const
IN num-var
IN (num-expr)

INK

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT X

הצהרה/פקודה

INK מגדירה את הצבע העליון בו מוצגים התוים, הנקודות המותוות וקווים ועקומות מצויירים.

כיצד להשתמש ב־INK

ניתן להשתמש ב־INK כפקודה ישירה אך בד"כ היא משמשת ליצירת הצהרה בתכנית. היא מלווה בערך מספרי, לדוגמה

```
7Ø INK x
```

הערך המלווה את INK מעוגל לשלם הקרוב ביותר ועשוי להמצא בתחום מØ עד 9. הצבעים העליונים הם כלהלן:

Ø שחור
1 כחול

2 אדום
3 סגול
4 ירוק
5 תכלת
6 צהוב
7 לבן
8 שקוף
9 צבע ניגודי שחור או לבן.

INK 8 מגדירה שהצבע הקיים נשאר ללא שינוי בכל נקודה על המסך עבורה הוגדר INK 8. INK 9 גורמת לצבע הדיו להיות שחור או לבן כך שיבלוט על רקע צבע הנייר (הרקע).

צבעי דיו מקיפים ומקומיים

כאשר INK יוצרת הצהרה לחוד, כמתואר לעיל, הצבע הוא מקיף וכל התצוגות הבאות יופיעו באותו צבע עליון. INK עשויה גם להיות מוטמעת (מוכנסת) בהצהרות תצוגה הנוצרות ע"י PRINT, INPUT, PLOT, DRAW ו־CIRCLE. INK באה אחרי מילת המפתח אך מקדימה את המידע או נתוני התצוגה; היא מלווה באותם ערכים ונקודה־פסיק, לדוגמה

```
6Ø CIRCLE INK 4 ; 128 , 88 , 87
```

ההשפעה של INK היא, איפוא, מקומית וישימה רק לתוים המוצגים, נקודות מותוות או קווים המצויירים ע"י הצהרת התצוגה, ובדוגמה זו ציור עיגול ירוק. לאחר מכן צבע הדיו חוזר לצבע המקיף או לברירת המחדל – שחור.

פורמט

INK int-num-expr [;]

INKEY$ מחרוזת מקש קלט

מיקום במקלדת

EXTEND MODE
N

פונקציה

INKEY$ משמשת לזיהוי לחיצת מקשים במקלדת.

כיצד להשתמש ב־INKEY$

INKEY$ אינה דורשת כל ארגומנט ומשמשת לרוב להקצאת תו עבור משתנה מחרוזת או לבדיקת תו מסויים, לדוגמה

```
7Ø LET a$ = INKEY$
13Ø IF INKEY$ = "N" THEN STOP
```

בעת הבצוע, INKEY$ מחזירה את התו הניתן ע"י המקש הנלחץ באותו רגע. אם לא נלחץ כל מקש, אזי INKEY$ מחזירה מחרוזת אפס (ריקה) (""). שים לב לכך ש־INKEY$ מבחינה בין אותיות גדולות וקטנות ותוים אחרים עיליים (SHIFT) או לא עיליים.

(העזר ב־IN לזיהוי מקש כלשהו ללא תווים מזהים).

שלא כמו INPUT, INKEY$ אינה מחכה אלא עוברת מייד להצהרה הבאה. לכן היא ממוקמת לרוב בתוך לולאה החוזרת ומתבצעת עד שהמקש הנדרש נלחץ.

דוגמה

שורה זו משהה את הפעולה עד אשר מקש y נלחץ (ללא SHIFT או CAPS LOCK).

```
6Ø IF INKEY$ <> "Y" THEN
  GOTO 6Ø
```

פורמט

INKEY$

INPUT

מיקום במקלדת

I

הצהרה/פקודה

INPUT מאפשרת הכנסת נתונים תוך כדי ריצת תכנית.

כיצד להשתמש ב־INPUT

INPUT יוצרת לרוב הצהרה בתכנית ומשמשת בצורה דומה ל־PRINT. בצורתה הפשוטה ביותר, היא מלווה במשתנה מספרי או מחרוזת, לדוגמה

```
6Ø INPUT x
9Ø INPUT a$
```

המחשב ממתין עד שיוכנס מספר או מחרוזת. הערך מוצג בתחילת השורה התחתונה תוך הקשתו פנימה. בלחיצת ENTER, הערך מוקצה למשתנה הנקרא והתכנית ממשיכה.

הצהרת INPUT עשויה לכלול יותר ממשתנה אחד והיא תציג תוים ליצירת סמן דרבון (prompt). זה מתבצע בדיוק באותה צורה כמו עם PRINT, תוך שימוש בסימני מרכאות הסוגרים את תווי סמן הדרבון וסימני נקודה־פסיק או פסיקים כנדרש להפרדה בין פריטים.

הצהרות תצוגה כמו INK, FLASH ו־PAPER ניתנות להטמעה, לדוגמה

```
8Ø INPUT  INK 2 ; "What is your
  name?" ; n$ , ("How old are you , "
  + n$ + "?") ; age
```

שים לב להבדלים הבאים לעומת PRINT. INPUT ממתינה כאשר היא מגיעה אל משתנה, ולכן כל המשתנים וביטויים (כמו אלה הכוללים n$ לעיל) העתידים להיכלל בדרבונים חייבים להיות בתוך סוגריים. התצוגה מתחילה בתחילת השורה התחתונה ומגולגלת מעלה אם יש יותר משורה אחת. ניתן להשתמש ב־AT בתוך הצהרת INPUT באותה צורה כמו עם PRINT. AT Ø,Ø מציגה בתחילת השורה מעל השורה התחתונה והתצוגה מגולגלת כלפי מעלה אם מוצגות יותר משתי שורות.

כיצד לעצור INPUT

אם INPUT מלווה בערך מספרי ומוקש STOP, התכנית תיעצר. עם משתנה מחרוזת, ניתן למחוק את סימן המרכאות הראשון ואז להכניס STOP כדי לעצור את התכנית.

שימוש ב־INPUT עם LINE

ניתן להשתמש ב־LINE INPUT עם משתני מחרוזת בלבד. בד"כ, INPUT עם משתנה מחרוזת גורמת להצגת צמד מרכאות. כאשר המחרוזת מוקשת פנימה, היא מוצגת בין המרכאות. כדי למחוק את המרכאות, העזר ב־LINE INPUT מלווה במשתנה המחרוזת. אם נדרש דרבון, הוא ממוקם בין INPUT ו־LINE, לדוגמה

```
7Ø INPUT "What is your
name?";LINE n$
```

פורמט

INPUT [prompt] [;] [,] ['] num-var
INPUT [prompt] [;] [,] ['] string-var
INPUT [prompt] [;] [,] ['] LINE string-var
[prompt] = [string-const] [(string-expr)] [AT int-num-expr, int-num-expr] [statement] [;] [,] [']

INT שלם

מיקום במקלדת

EXTEND MODE
R

פונקציה

INT משנה מספרים ממשיים (שאינם שלמים) למספרים שלמים.

כיצד להשתמש ב־INT

INT מלווה בערך מספרי, לדוגמה

```
70 LET x = INT y
```

ביטוי חייב להיות בתוך סוגריים. INT מחזירה את הערך כשהוא מעוגל למספר שלם.

דוגמה

הפקודה

```
PRINT INT 45.67 , INT -7.66
```

מציגה

```
45    -8
```

פורמט

INT num-const
INT num-var
INT (num-expr)

INVERSE

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT M

הצהרה/פקודה

INVERSE גורמת לצבעים להופיע הפוכים במיקומי תוים כך שצבע הדיו הופך לצבע הנייר ולהיפך.

כיצד להשתמש ב־INVERSE

INVERSE משמשת לרוב ליצירת הצהרה בתכנית. היא מלווה בערך מספרי, לדוגמה

```
70 INVERSE 1
```

הערך המלווה את INVERSE מעוגל לשלם הקרוב ביותר והוא עשוי להיות 0 או 1. INVERSE 1 גורם לכל התצוגות הבאות הנוצרות ע"י PRINT ו־INPUT להופיע בצבעים הפוכים. INVERSE 0 מחזירה את צבעי הדיו והנייר למצב הרגיל.

שים לב לכך שניתן להטמיע (להכניס) את INVERSE בתוך הצהרות תצוגה כפי שנעשה עם INK. ברם, כאשר פועלים עם CIRCLE ,PLOT או DRAW, INVERSE 1 גורמת לשורה או נקודה להיות מותווים בצבע הנייר כך שהם נעלמים.

פורמט

INVERSE int-num-expr

LEN אורך מחרוזת

מיקום במקלדת

EXTEND MODE
K

פונקציה

LEN נותנת אורך של מחרוזת.

כיצד להשתמש ב־LEN

LEN מלווה בערך מחרוזת, לדוגמה

```
50 LET x = LEN a$
```

בטוי חייב להיות בתוך סוגריים. LEN מחזירה את מספר התוים במחרוזת.

דוגמה

השורה הבאה

```
120 INPUT a$ : IF LEN a$ > 9
  THEN GOTO 120
```

מעבירה רק מחרוזות המכילות עד תשעה תוים.

פורמט

LEN string-const
LEN string-var
LEN (string-expr)

LET

מיקום במקלדת

L

הצהרה/פקודה

LET משמשת להקצאת ערך למשתנה. בבייסיק של סינקליר לא ניתן להשמיט את LET בהצהרת הקצאה.

כיצד להשתמש ב־LET

LET יוצרת לרוב הצהרה בתכנית אך עשויה לשמש גם כפקודה ישירה. היא מלווה במשתנה מספרי או מחרוזת, סימן שוויון, וערך. הערך יכול להיות מספרי או מחרוזת, בהתאם למשתנה שלפני LET, לדוגמה

```
60 LET x = x + 1
80 LET a$ = "Correct"
```

הערך מוקצה למשתנה.

שים לב שמשתנים פשוטים אינם מוגדרים עד שהם מקבלים ערכים ע"י LET, READ או INPUT. משתני מערך מצד שני מאותחלים ל־0 או מחרוזות אפס (ראה DIM).

פורמט

LET num-var = num-expr
LET string-var = string-expr

LINE

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT 3

ראה SAVE.

LIST

מיקום במקלדת

K

הצהרה/פקודה

LIST יוצרת רישום של התכנית הנמצאת בזכרון.

כיצד להשתמש ב־LIST

LIST משמשת לרוב כפקודה ישירה אך עשויה גם ליצור הצהרה בתכנית. כדי לרשום תכנית שלמה, משתמשים בה בנפרד. לאחר מתן הפקודה הישירה

```
LIST
```

מופיע העמוד הראשון ברישום והעמודים הבאים יגולגלו במעלה המסך בלחיצת מקש כלשהו פרט ל־N, מקש הרווחים, STOP או BREAK. ניתן גם ללוות את LIST במספר שורה, בצורת ערך מספרי, לדוגמה

```
LIST 100
```

הערך המלווה את LIST מעוגל לשלם הקרוב ביותר במידה ונדרש, והרישום מתחיל בשורה זאת. אם לא קיימת שורה עם מספר זה, הרישום מתחיל בשורה הבאה.

פורמט

LIST [int-num-expr]

LLIST רישום במדפסת

מיקום במקלדת

EXTEND MODE
V

פקודה/הצהרה

LLIST גורמת למדפסת מסוג סינקלייר להפיק תדפיס רישום של התכנית הנמצאת בזכרון.

כיצד להשתמש ב־LLIST

LLIST פועלת באותה צורה כמו LIST (ראה LIST לפרטים נוספים). שים לב שתצוגת המסך אינה משתנה עם הדפסת הרישום.

פורמט

LLIST [int-num-expr]

LN לוגריתם טבעי

מיקום במקלדת

EXTEND MODE
Z

פונקציה

LN נותנת את הלוגריתם הטבעי (לוגריתם על בסיס e) של ערך. היא פועלת כהפכית של EXP.

כיצד להשתמש ב־LN

LN מלווה בערך מספרי, לדוגמה

```
60 LET x = LN y
```

בטוי חייב להיות בתוך סוגריים. הערך המלווה את LN חייב להיות גדול מ־0. LN מחזירה את הלוגריתם הטבעי של ערך זה.

פורמט

LN num-const
LN num-var
LN (num-expr)

LOAD

מיקום במקלדת

J

פקודה/הצהרה

LOAD טוענת תכנית שלמה אל הזכרון מתוך סרט.

כיצד להשתמש ב־LOAD

LOAD משמשת לרוב כפקודה ישירה, אך היא עשויה ליצור הצהרה בתכנית כדי לטעון תכנית חדשה. LOAD מלווה בשם קובץ, שהוא משתנה מחרוזת באורך עד 10 תווים, לדוגמה

```
LOAD "file name"
```

בעת הבצוע, התכנית הנמצאת כעת בזכרון, וכל ערכי משתניה, נמחקים. הספקטרום מחפש אחר התכנית ששמה נקרא וטוען אותה לאחר שנמצאה. שים לב שהמחשב מבחין בין אותיות גדולות וקטנות בשמות תכניות.

אם בעקבות LOAD באה מחרוזת אפס, כמו בדוגמה זאת

```
LOAD ""
```

אזי הספקטרום טוען את התכנית השלמה הראשונה שהוא מוצא.

שים לב שהשימוש ב־LOAD שונה כאשר מחובר כונן זעיר. ראה מדריך כונן זעיר ומימשק ZX 1 לפרטים נוספים.

פורמט

LOAD string-expr

LOAD CODE

מיקום במקלדת

J
EXTEND MODE
I

פקודה/הצהרה

LOAD CODE משמשת לטעינת קטע בזכרון בנתונים שאוחסנו על סרט. הנתונים כוללים מערך של בתים, ואלה נשלחים אל מערך כתובות בזכרון. ניתן להשתמש ב־LOAD CODE לטעינת תצוגה, או לטעינת נתונים עבור גרפיקה בהגדרת משתמש, לדוגמה.

כיצד להשתמש ב־LOAD CODE

ניתן להשתמש ב־LOAD CODE כפקודה ישירה או ליצירת הצהרה בתכנית. LOAD מלווה בשם קובץ, שהוא ערך מחרוזת, ואחריו CODE, לדוגמה

```
LOAD "data" CODE
```

שם הקובץ אחרי LOAD הוא שם הנתונים הנטענים והוא כפוף לאותן הגבלות כמו שם תכנית (ראה LOAD). LOAD CODE מחפשת את שם הנתונים וכאשר הוא נמצא, מוצגים הבתים בלווית שם קובץ הנתונים. הספקטרום טוען את הבתים בזכרון באותן כתובות שמהן נשלחו לאחסון. בתהליך הטעינה עולים על כל הנתונים שהיו רשומים בזכרון מקודם.

ניתן גם ללוות את CODE באחד או שני ערכים מספריים, המופרדים בפסיק, לדוגמה

```
LOAD "picture" CODE
16384 , 6912
```

הערכים הבאים אחרי CODE מעוגלים לשלם הקרוב ביותר ואז מגדירים את כתובת ההתחלה (16384 שלעיל) בה יש לטעון את הנתונים הנקראים, ואת מספר הבתים (6912) אותם יש לשלוח לכתובות המתחילים באותה כתובת. אם המספר שגוי, ניתן דווח שגיאת טעינת סרט. אם אחרי CODE בא רק ערך אחד, הוא מגדיר את הכתובת הראשונה אשר ממנה והלאה ימוקמו כל הבתים.

הדוגמה שלעיל ניתנת לביצוע גם באמצעות מילות המפתח LOAD SCREEN$.

לפרטים נוספים על אחסון בתים, ראה SAVE CODE.

פורמט

LOAD string-expr CODE
[int-num-expr][,int-num-expr]

LOAD DATA

מיקום במקלדת

J
EXTEND MODE
D

הצהרה/פקודה

LOAD DATA משמשת לטעינת מערכים מסרט. המערכים מוקלטים בעזרת SAVE DATA.

כיצד להשתמש ב־LOAD DATA

ניתן להשתמש ב־LOAD DATA ליצירת הצהרה בתכנית או כפקודה ישירה. LOAD מלווה בתחילה בשם קובץ, שהוא משתנה מחרוזת, ואחריו DATA בתוספת אות או אות וסימן $, ולבסוף בצמד סוגריים ריקים, לדוגמה

```
270 LOAD "numbers" DATA n()
300 LOAD "names" DATA n$ ()
```

שם הקובץ המלווה את LOAD הוא השם הניתן למערך על הסרט וכפוף לאותן הגבלות כמו שמות תכניות המשמשים ב־LOAD. האות או אות וסימן $ בעקבות DATA היא השם הניתן למערך בתכנית כאשר הוא נטען ומופעל.

בעת הבצוע, הספקטרום מחפש את שם המערך על הסרט. כאשר השם נמצא, מופיעה ההודעה על מערך מספרי או מערך תוים, בלווית השם, והמערך נטען.

כל מערך הנמצא מקודם בזכרון ונושא אותו שם אות (n או n$ לעיל) נמחק, ונוצר מערך חדש באותו שם אות כשכל ערכיו מאוחסנים בסרט. שים לב שבמערכי תוים, כל משתנה מחרוזת הנמצא אותו זמן בזכרון ונושא אותו שם אות נמחק אף הוא.

פורמט

LOAD string-expr DATA letter [$]()

LOAD SCREEN$

מיקום במקלדת

J
EXTEND MODE
SYMBOL SHIFT K

הצהרה/פקודה

LOAD SCREEN$ מאפשרת טעינת תצוגת המסך ישירות מהסרט. היא שולחת נתונים מהסרט לקטע בזכרון אשר מבקר את תצוגת המסך במטרה להפיק את התמונה.

כיצד להשתמש ב־LOAD SCREEN$

ניתן להשתמש ב־LOAD SCREEN$ ליצירת הצהרה בתכנית או כפקודה ישירה. LOAD מלווה בשם קובץ, שהוא ערך מחרוזת, ואחריו SCREEN$, לדוגמה

```
LOAD "picture" SCREEN$
```

שם הקובץ הבא אחרי LOAD הוא השם הניתן לנתוני המסך שעל הסרט, כפוף לאותן הגבלות כמו שמות תכניות הנמצאים בשימוש עם LOAD. הספקטרום מחפש את הנתונים ששמם נקרא וכאשר מוצא אותם, הם נטענים ראשית לקובץ התצוגה ואח"כ לקטע המאפיינים בזכרון. התמונה נבנית לאט בצבעי הדיו והנייר הקיימים ואז נוספים המאפיינים (צבעים אמיתיים וכן הלאה).

לפרטים נוספים על אחסון נתוני מסך, ראה SAVE SCREEN$.

פורמט

```
LOAD string-expr SCREEN$
```

LPRINT הדפסה במדפסת

מיקום במקלדת

EXTEND MODE
C

הצהרה/פקודה

LPRINT גורמת למדפסות מסוג סינקלייר להדפיס פריט מידע באותו אופן בו PRINT גורמת לפריט להופיע על המסך.

כיצד להשתמש ב־LPRINT

LPRINT עשויה ליצור הצהרה בתכנית או פקודה ישירה. היא מלווה בפריטי מידע היכולים להיות מופרדים בנקודה־פסיק, פסיקים, או גרשיים, לדוגמה

```
60 LPRINT "Number" ; x'
  "Name" ; n$ , "Age" ; a
```

כאשר הפלט עובר למדפסת, הנתונים מודפסים באותו פורמט כפי ש־PRINT היתה מציגה אותם על המסך. הצהרת או פקודת LPRINT עשויה לכלול גם הצהרות TAB, בקרי CHR$ מסוימים, הצהרות INVERSE ו־OVER וקודי בקרה עם אותה השפעה כמו PRINT.

ניתן לכלול גם הצהרת AT, אך מתעלמים ממספר השורה ופריט המידע מודפס בנקודת העמודה הנתונה באותה השורה.

פורמט

```
LPRINT [TAB int-num-expr;)]
[AT int-num-expr,int-num-expr;]
[CHR$(int-num-expr);][statement;]
[num-expr][string-expr][;][,][']
```

MERGE

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT T

הצהרה/פקודה

MERGE מאפשרת מיזוג של שתי תכניות.

כיצד להשתמש ב־MERGE

ניתן להשתמש ב־MERGE ליצירת הצהרה בתכנית או כפקודה ישירה. היא מלווה בשם קובץ בצורת ערך מחרוזת, לדוגמה

```
500 MERGE "prog 2"
```

שם הקובץ הבא אחרי MERGE הוא שם התכנית שתתמזג עם התכנית הקיימת בזכרון. שם זה כפוף לאותן מגבלות שם תכנית הקיימות בשימוש עם LOAD. MERGE טוענת את התכנית החדשה בלי למחוק תחילה את התכנית הקיימת. ברם, התכנית החדשה נכתבת על גבי התכנית הקיימת במקומות בהם קיימים אותם מספרי שורות ושמות משתנים כמו בתכנית הקיימת.

פורמט

```
MERGE string-expr
```

MOVE

פקודת טיפול בקבצים בכונן זעיר. ראה מדריך כונן זעיר ומימשק 1.

NEW

מיקום במקלדת

A

הצהרה/פקודה

NEW מנקה את מרחב זכרון מערכת בייסיק (כל התחום עד RAMTOP) ומוחקת כל תכנית הקיימת בחלק זה של הזכרון.

כיצד להשתמש ב־NEW

NEW משמשת לרוב כפקודה ישירה אך עשוייה גם ליצור הצהרה בתכנית. היא פועלת בנפרד. בעת הביצוע, נמחקים התכנית והמשתנים. הזכרון נמחק עד למחיצת RAMTOP כך שתווי גרפיקה בהגדרת משתמש, אשר מאוחסנים מעל RAMTOP, אינם מושפעים.

פורמט

```
NEW
```

NEXT

מיקום במקלדת

N

הצהרה/פקודה

NEXT נמצאת תמיד בשימוש יחד עם FOR ליצירת לולאת FOR NEXT.

כיצד להשתמש ב־NEXT

NEXT משמשת לרוב ליצירת הצהרה בתכנית להשלמת לולאת FOT NEXT היא מלווה באות שהיא משתנה הבקרה בלולאה, לדוגמה

```
90 NEXT a
```

בבייסיק של סינקלייר, חייבים לכלול את משתנה הבקרה.

ראה FOR לפרטים נוספים על לולאות FOR NEXT.

פורמט

```
NEXT letter
```

NOT

מיקום במקלדת

SYMBOL SHIFT S

אופרטור לוגי/פונקציה

NOT משמש להיפוך מצב אמת של תנאי כך שתנאי שקרי הופך לאמת ולהיפך.

כיצד להשתמש ב־NOT

NOT מלווה בתנאי או בערך מספרי, לדוגמה

```
90 IF NOT x = y + z THEN PRINT
  "Wrong"
90 LET correct = x = y + z : IF NOT
  correct THEN PRINT "Wrong"
```

כאשר NOT מלווה בתנאי (x = y + z לעיל), הספקטרום מקצה ראשית ערך של 1 לתנאי אם הוא אמת ו־Ø אם הוא שקרי. NOT משמש, איפוא, כפונקציה להיפוך הערך שנוצר, כך שניתן לבדוק את התנאי ההפכי. שים לב שתנאי חייב להיות בתוך סוגריים אם הוא מכיל AND או OR.

אם NOT מלווה בערך מספרי, היא מחזירה Ø אם הערך המלווה שונה מאפס ו־1 אם הערך המלווה הוא Ø. לכן בדוגמאות שלעיל, הספקטרום מדפיס "Wrong" אם x <> y + z או אם correct הוא בעל ערך Ø.

פורמט

NOT cond
NOT num-expr

OPEN

פקודת טיפול בקבצים בכונן זעיר. ראה מדריך כונן זעיר ומימשק 1.

OR

מיקום במקלדת

SYMBOL SHIFT U

אופרטור לוגי/פונקציה

OR פועלת כאופרטור לוגי לבדיקת מצב אמת של צירוף תנאים. אם תנאי אחד או יותר הם אמת, אזי כל הצירוף הוא אמת. OR פועלת גם כפונקציה לבצוע פעולות בינאריות על שני ערכים מספריים.

כיצד להשתמש ב־OR

בתור אופרטור לוגי, OR מקשרת בין שני תנאים בהצהרה בה יש לבדוק את אמיתות כל הצירוף, לדוגמה

```
7Ø IF INKEY$ = "N" OR
  INKEY$ = "n" THEN STOP
```

אם אחד או שני התנאים הם אמת, אזי הצירוף כולו אמת. בשורה שלעיל, אחד מהתנאים (INKEY$ = "N" ו־INKEY$ = "n") הופך לאמת ברגע שמקש N נלחץ, ללא קשר למצב מקשי CAPS SHIFT או CAPS LOCK. כל הצירוף הופך לאמת והתכנית נעצרת.

OR בתור פונקציה

הספקטרום +ZX מקצה ערך מספרי של 1 לתנאי אמת ו־Ø לתנאי שקרי. הוא מכיר כל ערך שונה מאפס כאמת ו־Ø כשקר. ניתן איפוא להקדים או ללוות את OR בערך מספרי, לדוגמה

```
4Ø LET x = y OR z
```

המשתנה x מקבל ערך של 1 אם z שונה מאפס או תנאי אמת, או ערך של y אם z שווה Ø או תנאי שקרי.

הדבר שימושי באריתמטיקה. בדוגמה הבאה, מחיר הכרטיס (fare) מחולק ב־2 אם הגיל (age) קטן מ־14.

```
6Ø PRINT fare * (Ø.5 OR age> 13)
```

אם הגיל קטן מ־14, התנאי age > 13 הוא שקרי, ולכן מחיר הכרטיס מוכפל ב־Ø.5. אם age > 13 נכון, אזי המחיר מוכפל ב־1.

שים לב שהספקטרום אינו מחשב צירופים של ערכים מספריים בהתאם לטבלאות אמת תקניות.

פורמט

cond OR cond
num-expr OR num-expr

OUT

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT O

הצהרה/פקודה

OUT שולחת בית לכתיבת שער (Port) נתונה במטרה להניע התקן פלט.

כיצד להשתמש ב־OUT

ניתן להשתמש ב־OUT ליצירת הצהרה בתכנית או כפקודה ישירה. היא מלווה בשני ערכים מספריים, המופרדים בפסיק, לדוגמה

```
4Ø OUT 254 , 3
```

שני הערכים מעוגלים לשלם הקרוב ביותר. הערך הראשון (254 לעיל) עשוי להיות בתחום בין Ø ו־65535 והוא כתובת השער. הערך השני (3) עשוי להמצא בין Ø ו־255 והוא הבית העתיד להישלח לאותה כתובת.

סיביות Ø עד 2 של בית הפלט לכתובת השער 254 קובעות את צבע הגבול; הדוגמה שלעיל מחזירה איפוא צבע גבול סגול. סיבית 3 בכתובת זו מפעילה את שקע MIC וסיבית 4 את הרמקול. כתובת שער 251 מפעילה את המדפסת, ושערים 254, 247 ו־239 משמשים עבור רכיבים היקפיים אחרים.

פורמט

OUT int-num-expr , int-num-expr

OVER

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT N

הצהרה/פקודה

OVER משמשת להדפסת תו אחד על גבי תו אחר. ניתן גם להתוות בעזרתה נקודות או לצייר קוים ועקומות בצבע נייר במקום בצבע דיו.

כיצד להשתמש ב־OVER

OVER משמשת בד"כ ליצירת הצהרה בתכנית. היא מלווה בערך מספרי, לדוגמה

```
8Ø OVER 1
```

הערך הבא אחרי OVER מעוגל לשלם הקרוב ביותר ועשוי להיות איפוא Ø או 1. OVER Ø, שהוא ברירת המחדל (קבוע מראש), גורמת לכל תו למחוק תו קודם שהיה באותו מיקום ולהחליפו. OVER 1 גורמת לשילוב של כל שני תוים המוצגים באותו מיקום תו.

OVER ניתנת להטמעה (הכנסה) בתוך הצהרת PRINT או INPUT באותו אופן כמו INK כך שהיא משפיעה רק על תוים המוצגים ע"י ההצהרה.

ההצהרה הבאה, למשל, מדגישה מילה בקו

```
6Ø PRINT AT 11 , 15 ; "YES" ;
  OVER 1 ; AT 11 , 15 ; "___"
```

ברם, שים לב שהתוים משולבים כך שבמקום בו צבעי הדיו חופפים, ניתן צבע הנייר.

OVER במצב הפרדה גבוהה

OVER ניתן לשימוש עם PLOT, DRAW ו־CIRCLE. ללא OVER, קוים ועקומות יכולים לחפוף זה את זה, אך הם חייבים להיות באותו צבע דיו אחרת צבע הדיו בכל מיקום התו ישתנה במקומות החצייה. אם מופעלת OVER 1, קוים או עקומות יוצרים את צבע הנייר במקומות החפיפה או פגישת תוים אחרים.

התוויית נקודות או ציור קווים ועקומות שנית באותו מקום כאשר מופעל OVER 1 גורמת להעלמותם מהמסך.

פורמט

OVER int-num-expr

PAPER

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT C

הצהרה/פקודה

PAPER משמשת לבחירת צבע הנייר או הרקע עבור תצוגת המסך. זה עשוי להיות צבע הרקע על פני כל שטח התצוגה, או הצבע מאחורי תוים, נקודות, או קוים בודדים המופיעים במיקומי תוים בודדים.

כיצד להשתמש ב־PAPER

ניתן להשתמש ב־PAPER ליצירת הצהרה בתכנית או כפקודה ישירה. הוא מלווה בערך מספרי, לדוגמה

```
80 PAPER x
```

הערך המלווה את PAPER מעוגל לשלם הקרוב ביותר ועשוי לנוע בין 0 עד 9. צבעי הנייר הניתנים זהים לאלה הניתנים ע"י INK. צבעי נייר עשויים גם להיות מקיפים או מקומיים ע"י הטמעתם (הכנסתם) בהצהרות תצוגה בדיוק כמו עם צבעי דיו. ראה INK לפרטים נוספים.

כאשר תוים מודפסים לאחר הצהרת PAPER, מקיפה או מקומית, הרקע סביב כל מיקום התו שעליו פעלו משתנה לצבע הנבחר. הדבר אמור גם לגבי מצב בו הותוו נקודות או צויירו קוים ועקומות בעזרת הצהרת PAPER מוטמעת, אך לא אחרי פקודה או הצהרה מקיפות.

כדי ליצור רקע צבעוני על כל שטח התצוגה, הכרחי להשתמש ב־CLS לאחר הצהרת PAPER. כל התצוגה עוברת לצבע זה, אשר נשאר גם אח"כ צבע הרקע הכללי.

פורמט

PAPER int-num-expr [;]

PAUSE

מיקום במקלדת

M

הצהרה/פקודה

ניתן להשתמש ב־PAUSE לצורך השהיית תכנית לפרק זמן מוגדר או בלתי מוגבל.

כיצד להשתמש ב־PAUSE

PAUSE משמשת בד"כ ליצירת הצהרה בתכנית. היא מלווה בערך מספרי, לדוגמה

```
130 PAUSE 100
```

הערך המלווה את PAUSE מעוגל לשלם הקרוב ביותר ועשוי להמצא בתחום בין 0 ר־65535.

הערך מגדיר את ההשהייה כמספר מסגרות תמונת הטלביזיה, כך שערך של 50 יוצר השהייה בת 1 שנייה בישראל, בריטניה ואירופה בהן תדר המסגרת הוא 50 הרץ.

ברם, שים לב שניתן להפסיק כל השהייה ע"י לחיצת מקש כלשהו ר PAUSE 0 נותנת השהייה בלתי מוגבלת הנמשכת עד ללחיצת מקש כלשהו.

פורמט

PAUSE int-num-expr

PEEK

מיקום במקלדת

EXTEND MODE
O

פונקציה

PEEK נותנת את ערך הבית המאוחסן בכתובת מסוימת בזכרון.

כיצד להשתמש ב־PEEK

PEEK מלוה בערך מספרי, לדוגמה

```
80   LET x = PEEK(256 * y)
```

שים לב שבטוי חייב להיות בתוך סוגריים. הערך המלווה את PEEK מעוגל לשלם הקרוב ביותר אם נדרש, ויכול לנוע בין 0 עד 65535 עבור כתובת בזכרון. PEEK מחזירה את ערך הבית (מספר בין 0 ל־255) בכתובת הנתונה.

דוגמה

מספר מסגרות תצוגת הטלביזיה שהתרחשו מאז הופעל הספקטרום לאחרונה מאוחסן בכתובות 23672 עד 23674. הצצה לכתובות אלה, תוך כדי יצירת המסגרות בקצב קבוע נותנת שיטה למדידת זמן. השורה הבאה מציגה את הזמן בשניות שעבר מאז הופעל הספקטרום לאחרונה (פחות הזמן שנוצל להפקת קול והפעלת רכיבים היקפיים כרשמקול ומדפסת).

```
10 PRINT (PEEK 23672 + 256 *
  PEEK 23673 + 65536 * PEEK
  23674)/50
```

הערה — אם תדר הרשת הוא 60 הרץ ולא 50 הרץ (תקן בריטי), שנה 50 ל־60.

פורמט

PEEK int-num-const
PEEK int-num-var
PEEK (int-num-expr)

PI

מיקום במקלדת

EXTEND MODE
M

פונקציה

PI נותנת את הערך של פי (π) לשימוש בחישובים. PI הוא היחס בין היקף העיגול לקוטרו.

כיצד להשתמש ב־PI

PI אינה דורשת כל ערכים או משתנים כאשר היא מופעלת בהצהרה או פקודה, לדוגמה

```
DRAW 255 , 0 , —PI
```

PI מחזירה ערך של 3.1415927 כך שהפקודה שלעיל מציירת חצי מעגל גדול על המסך.

פורמט

PI

PLOT

מיקום במקלדת

Q

הצהרה/פקודה

PLOT משמשת בגרפיקה בהפרדה גבוהה להתוויית פיקסל או נקודת צבע במיקום מסויים על המסך.

כיצד להשתמש ב־PLOT

PLOT משמשת ליצירת הצהרה בתכנית או כפקודה. בד"כ היא מלווה

בשני ערכים מספריים המופרדים בפסיק, לדוגמה

```
50 PLOT 128 , 87
```

שני הערכים המלווים את PLOT מעוגלים לשלמים במידת הצורך. הערך הראשון עשוי לנוע בין 0 ל־255 ומגדיר את הקואורדינטה האפקית של הנקודה על המסך. הערך השני עשוי לנוע בין 0 ל־175 ומגדיר קואורדינטה אנכית. הפיקסל יותווה בד"כ בצבע הדיו הקיים בנקודה המוגדרת – בדוגמה שלעיל, במרכז המסך.

שים לב להשפעות הבאות של הצהרות או פקודות צבע על PLOT. אחרי OVER 1, כל נקודה קיימת באותו מיקום משנה את צבעה לצבע הנייר. לאחר INVERSE 1, הנקודה מותווה בצבע הנייר הקיים. לאחר BRIGHT 1 או FLASH 1, כל מיקום התו על המסך בהפרדה נמוכה שעליו מותווה הפיקסל יהיה בהיר או מהבהב.

ארבע מילות מפתח אלה ו־INK ניתנות גם להטמעה בתוך הצהרת PLOT באותו אופן כמו עם PRINT, לדוגמה

```
160 PLOT INK 2 ; x , y
```

השפעתן זהה אך היא מקומית ומוגבלת לפיקסל המותווה ע"י ההצהרה. אם PAPER מוטמעת בהצהרת PLOT, צבע הנייר של כל מיקום התו סביב הפיקסל משתנה לצבע הנתון.

שים לב ש־PLOT גם מגדירה את נקודת ההתחלה של הצהרת DRAW הבאה.

פורמט

PLOT [statement ;]
int-num-expr , int-num-expr

POINT

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT 8

פונקציה

POINT משמשת לבירור האם הצבע בנקודה מסויימת על המסך בהפרדה גבוהה הוא צבע דיו או צבע נייר. POINT אינה בודקת את הצבע עצמו.

כיצד להשתמש ב־POINT

POINT מלווה בשני ערכים מספריים המופרדים בפסיק ונמצאים בתוך סוגריים, לדוגמה

```
240 IF POINT (x , y) = 1 THEN
  GOSUB 600
```

שני הערכים המלווים את POINT מעוגלים לשלמים במידת הצורך. הערך הראשון עשוי לנוע בין 0 ל־255 ומגדיר את הקואורדינטה האפקית של פיקסל על המסך. הערך השני עשוי לנוע בין 0 ל־175 ומגדיר קואורדינטה אנכית. POINT מחזירה 1 אם הפיקסל בנקודה המוגדרת הוא צבע דיו או 0 אם הוא צבע נייר.

פורמט

POINT (int-num-expr , int-num-expr)

POKE

מיקום במקלדת

O

הצהרה/פקודה

POKE משמשת לשינוי ערך הבית בכתובת מסויימת בזכרון. ערכים בד"כ "נדחפים" למקומות בזכרון במטרה ליצור פעולות שאינן ניתנות לבצוע ע"י מילות מפתח בבייסיק.

כיצד להשתמש ב־POKE

POKE משמשת ליצירת הצהרה בתכנית או פקודה. היא מלווה בשני ערכים מספריים המופרדים בפסיק, לדוגמה

```
POKE 23609 , 255
```

שני הערכים המלווים את POKE מעוגלים לשלמים הקרובים ביותר במידת הצורך. הערך הראשון עשוי לנוע בין 16384 עד 65535 ומהווה כתובת בזכרון ראם. הערך השני עשוי לנוע בין 0 עד 255 ומהווה את הביית האמור להכתב בכתובת המוגדרת.

בדוגמה שלעיל, 255 "נדחף" לכתובת 23609, אשר מבקרת את הקול כאשר נלחץ מקש. ערך של 255 יוצר צפצוף ארוך במקום ה"קליק" הרגיל, וערכים אחרים יוצרים צפצוף קצר יותר.

פורמט

POKE int-num-expr ,
int-num-expr

PRINT

מיקום במקלדת

P

הצהרה/פקודה

PRINT מציגה מידע על המסך. המידע עשוי להיות כל תו בודד או רצף תוים. הצהרת PRINT עשויה לכלול מילות מפתח אחרות להגדרת מיקום וצבע המידע.

כיצד להשתמש ב־PRINT

ניתן להשתמש ב־PRINT לחוד או כשהיא מלווה במידע. מידע זה עשוי להיות בצורת כל בטוי מספרי או מחרוזת, או תערובת שלהם. כאשר משתמשים ב־PRINT עם נתונים, שני פריטים נפרדים או יותר חייבים להיות מופרדים בנקודה־פסיק, פסיק או גרש.

ניתן להכניס מילות מפתח מסויימות אחרות בכל סדר שהוא בין PRINT והנתונים, בתנאי שכל הצהרה הנוצרת ע"י מילות המפתח מסתיימת בנקודה־פסיק. מילות מפתח אלה הן CHR$, TAB, AT, INK, PAPER, FLASH, BRIGHT, INVERSE ו־OVER.

PRINT עם מחרוזות

PRINT לחוד או בלוויית מחרוזת אפס ("") מציגה שורה ריקה ומעבירה את הסמן לתחילת השורה הבאה.

PRINT מלווה בקבוע מחרוזת (תוים כלשהם בין מרכאות כפולות) מציגה את התוים כפי שהם מופיעים בין סימני המרכאות. הפקודה

```
PRINT "3/542/76/21"
```

לדוגמה, מציגה

```
3/542/76/21
```

PRINT מלווה במשתנה מחרוזת או בטוי מציגה את המחרוזת או מחרוזות שהם מייצגים.

PRINT עם מספרים

PRINT מלווה בבטוי מספרי כלשהו מציגה את ערך הבטוי. מספרים מוצגים בפורמט עשרוני עם עד ל־8 ספרות משמעותיות וללא אפסים עוקבים אחרי הנקודה העשרונית. מספרים גדולים מאד וקטנים מאד מוצגים בפורמט מדעי מקוצר, כשני מספרים המופרדים באות E. זה מציין

מספר שבו החלק הראשון (המנטיסה) מועלית בחזקת החלק השני (המעריך). הפקודה

```
PRINT 3/542/76/21
```

לדוגמה, מציגה

```
3.4680798 E −6
```

שנוי פורמט PRINT בעזרת סימני פיסוק

PRINT מלווה בפריטי מידע המופרדים בנקודה־פסיק מציגה את הפריטים כשהם זה בצד זה ללא רווח. הפקודה

```
PRINT 1 ; 2 ; 3
```

מציגה

```
123
```

PRINT מלווה בפריטי מידע המופרדים בפסיק מציגה כל פריט בתחילת או באמצע השורה בהתאם למיקום הפריט הראשון. הפקודה

```
PRINT 1 , 2 , 3
```

מציגה

```
1               2
3
```

PRINT מלווה בפריטי מידע המופרדים בגרש מציגה את הפריט שאחרי הגרש בתחילת השורה הבאה. הפקודה

```
PRINT 1 ' 2 ' 3
```

מציגה

```
1
2
3
```

אם הצהרת או פקודת PRINT מסתיימת בנקודה־פסיק, פסיק או גרש, אזי הפריט שיוצג ע"י הצהרת PRINT הבאה יושפע באותו אופן.

PRINT ומילות מפתח אחרות

ניתן ללוות את PRINT ב־TAB, ערך מספרי, נקודה־פסיק ואחריהם פריט מידע, לדוגמה

```
60 PRINT TAB x ; a$
```

הערך הבא אחרי TAB (x שלעיל) מעוגל לשלם הקרוב ביותר במידת הצורך ואז מחולק ב־32 והשארית מוחזרת בצורת ערך הנע בין 0 ו־31. פריט המידע מוצג על המסך במספר עמודה זה באותה השורה או בשורה הבאה.

ניתן ללוות את PRINT ב־AT ושני ערכים מספריים המופרדים בפסיק, נקודה־פסיק ופריט מידע, לדוגמה

```
50 PRINT AT l , c ; "Data"
```

הערך הראשון (l שלעיל) עשוי לנוע בין 0 ו־21 ומגדיר את מספר השורה בה יוצג המידע. הערך השני (c) עשוי לנוע בין 0 ו־31 ומגדיר את מספר העמודה בה יוצג התו הראשון או הספרה הראשונה של פריט המידע. ערכים לא שלמים מתקבלים ומעוגלים לשלם הקרוב ביותר. הפקודה PRINT AT 11 , 16 ; "*" מציגה כוכב במרכז המסך.

ניתן גם ללוות את PRINT באחת או יותר פונקציות CHR$. ראה CHR$ לפרטים נוספים.

PRINT ומילות מפתח לצבע

התצוגה המופקת ע"י PRINT מושפעת מהצהרות או פקודות צבע הניתנות ע"י INK, PAPER, FLASH, BRIGHT, INVERSE ו־OVER הנמצאות אותו זמן בפעולה. ניתן גם ללוות את PRINT באחת או יותר משש הצהרות אלה כשכל אחת מלווה בנקודה־פסיק לפני פריט המידע, לדוגמה

```
50 PRINT AT 11 , 16 ; INK 2 ;
  FLASH 1 ; "*"
```

פריט המידע מוצג עם המאפיינים המוגדרים ע"י מילות המפתח לצבע. מאפיינים אלה הם מקומיים וישימים רק לפריט המוצג. לאחר בצוע הצהרת PRINT, הם חוזרים לערכיהם בברירת המחדל או ערכים מקיפים שהוכרזו מקודם. PRINT כפופה גם לקודי בקרת צבע מקומיים המוכנסים עם המידע (ראה עמ' 33).

פורמט

```
PRINT [TAB int-num-expr ; ]
[AT int-num-expr , int-num-expr ; ]
[CHR$ (int-num-expr) ; ][statement ; ]
[num-expr][string-expr][ ; ][ , ][ ' ]
```

RANDOMIZE

מיקום במקלדת

T

הצהרה/פקודה

RANDOMIZE, המופיעה על המקלדת כ־RAND, משמשת בהקשר עם RND ליצירת רצפי מספרים שהם אקראיים או צפויים מראש.

כיצד להשתמש ב־RANDOMIZE

RANDOMIZE משמשת ליצירת הצהרה בתכנית או כפקודה. ניתן ללוותה, כאופציה, בערך מספרי, לדוגמה

```
RANDOMIZE 1
10 RANDOMIZE
```

הערך המלווה את RANDOMIZE מעוגל לשלם הקרוב ביותר במידת הצורך ועשוי לנוע בין 0 עד 65535. ערך גדול מ־0 קובע את משתנה המערכת SEED לערך זה, הבא אחרי רצף המספרים הנוצר תמיד ע"י RND (ראה עמ' 48 למידע על משתני מערכת). הרצף המעשי לא תלוי בערך של RANDOMIZE.

אם RANDOMIZE מלווה ב־0 או ללא ערך, אזי SEED מקבל ערך של משתנה מערכת אחר הקרוי FRAMES, אשר מונה את המסגרות המוצגות על מסך הטלביזיה מאז הופעל הספקטרום לאחרונה. מכיון ש־SEED משתנה 50 או 60 פעם בשנייה, רצף המספרים הנוצר ע"י RND לאחר RANDOMIZE או RANDOMIZE 0 הוא אקראי ביותר.

אם RANDOMIZE אינו בשימוש, RND יוצר את אותו רצף מספרים מזמן הפעלת המחשב ואחרי שימוש בכפתור RESET או פקודת NEW.

פורמט

```
RANDOMIZE [int-num-expr]
```

READ

מיקום במקלדת

EXTEND MODE
A

הצהרה/פקודה

READ משמשת בהקשר עם DATA להקצאת ערכים עבור משתנים תוך שימוש בערכים שבהצהרת DATA.

כיצד להשתמש ב־READ

READ משמשת בד"כ ליצירת הצהרה בתכנית. היא מלווה באחד או יותר

ערכים מספריים או משתני מחרוזות המופרדים בפסיקים, לדוגמה

```
20 READ a$ , x
```

כאשר READ מתבצעת בראשונה, היא לוקחת את אותו מספר ערכים כמספר המשתנים מתחילת רשימת DATA הראשונה ומקצה לפי הסדר את הערכים למשתנים. כאשר READ מתבצעת בשנית, המערך הבא של ערכי DATA מוקצה למשתנים הנקראים בהצהרת READ וכן הלאה.

לפרטים נוספים, ראה DATA.

פורמט

READ num-var [, num-var]
[, string-var]
READ string-var [, num-var]
[, string-var]

REM הערה

מיקום במקלדת

E

הצהרה

REM משמשת לכתיבת הערות או תזכורות בתכנית. אלה עשויות להיות הכותרת ושם יוצר התכנית, והסברים לשורות בתכנית כמו מטרת משתנה מסוים. ההערות לא משתתפות בריצת התכנית וניתן לראותן רק ברישום.

כיצד להשתמש ב־REM

REM יוצרת שורה משל עצמה בתכנית או מהווה הצהרה אחרונה בשורה. היא מלווה בהערה כלשהי הניתנת להקשה לפי הצורך, לדוגמה

```
8Ø INPUT n$ : REM n$ is name
```

כאשר המחשב נתקל ב־REM, הוא מתעלם מכל דבר הבא אחרי REM באותה שורה.

פורמט

REM תוים כלשהם

RESTORE

מיקום במקלדת

EXTEND MODE
S

הצהרה/פקודה

RESTORE משמשת בהקשר עם READ ו־DATA כדי ש־READ תיקח ערכים מהצהרת DATA מסוימת במקום הצהרת DATA הראשונה או שנייה בתכנית.

כיצד להשתמש ב־RESTORE

RESTORE יוצרת בד"כ הצהרה בתכנית. היא מלווה לפי בחירה בערך מספרי, לדוגמה

```
16Ø RESTORE 8ØØ
```

הערך הבא אחרי RESTORE מעוגל לשלם הקרוב ביותר במידת הצורך, ואז עליו להיות מספר שורה בתכנית אשר מכילה הצהרת DATA. לאחר RESTORE, הצהרת READ הבאה תקצה את ערכי הצהרת DATA הנ"ל. אם השורה הממוספרת אינה קיימת או שאינה מכילה הצהרת DATA, אזי READ תפנה להצהרת DATA הבאה אחרי שורה זו.

אם RESTORE מלווה ב־Ø או ללא ערך מלווה, אזי פקודת READ הבאה תפנה להצהרת DATA הראשונה בתכנית.

פורמט

RESTORE [int-num-expr]

RETURN

מיקום במקלדת

Y

הצהרה/פקודה

RETURN משמשת לסיום שיגרה והחזרת המחשב לתכנית הראשית או שיגרה קודמת.

כיצד להשתמש ב־RETURN

RETURN משמשת בד"כ ליצירת הצהרה בתכנית. היא מופיעה לחוד בסיום השגרה, לדוגמה

```
1Ø8Ø RETURN
```

בעת הבצוע, התכנית מסתעפת להצהרה הבאה אחרי הצהרת GOSUB האחרונה שבוצעה.

ראה GOSUB לפרטים נוספים.

פורמט

RETURN

RND מספר אקראי

מיקום במקלדת

EXTEND MODE
T

פונקציה

RND משמשת ליצירת מספר אקראי.

כיצד להשתמש ב־RND

RND מופיעה לחוד בהצהרה או פקודה, לדוגמה

```
6Ø LET x = RND
```

RND מחזירה מספר אקראי קטן מ־1 וגדול או שווה ל־Ø.

כאשר הספקטרום מופעל או מאותחל, או שהוקש NEW, יוחזרו מספרים ע"י RND באותו סדר. הרצף נוצר ע"י שימוש בחזקות של 75 (75, 75*75, 75*75*75 וכן הלאה), חלוקת כל חזקה ב־65537 וניצול השארית בלבד ע"י החסרת 1 ממנה וחלוקת התוצאה ב־65536.

אם נדרש רצף אקראי יותר או רצף קבוע אחר, השתמש ב־RANDOMIZE לפני RND.

מספרים שלמים אקראיים

הצהרות ופקודות רבות בספקטרום, כמו INK ו־CHR$, מעגלות מספרים לשלם הקרוב ביותר וניתן להשתמש בהם ישירות עם RND. לדוגמה, INK RND*7 יוצרת צבע דיו אקראי.

כדי לקבל מספר שלם אקראי מ־1 עד x ניתן להשתמש ב־INT(RND*x)+1.

ליצירת מספר שלם אקראי מ־Ø עד x, השתמש ב־INT (RND * x+Ø.5).

פורמט

RND

RUN

מיקום במקלדת

R

הצהרה/פקודה

RUN גורמת לתכנית לרוץ, בד"כ מהשורה הראשונה.

כיצד להשתמש ב־RUN

ניתן להשתמש ב־RUN כפקודה ישירה או ליצירת הצהרה בתכנית. היא מלווה, לפי בחירה, בערך מספרי, לדוגמה

```
RUN 5Ø
```

אם אין ערך אחרי RUN, התכנית רצה מהשורה הראשונה. אם נכלל ערך מסוים, הוא מעוגל לשלם הקרוב ביותר במידת הצורך והתכנית רצה

משורה זאת. אם השורה לא קיימת, התכנית תרוץ מהשורה הבאה בתכנית. שים לב ש־RUN מבצעת CLEAR לפני הרצת התכנית, כך שערכי המשתנים נמחקים. כדי למנוע זאת, השתמש ב־GOTO מלווה במספר שורה.

אם תכנית נשמרה באמצעות LINE, היא תרוץ אוטומטית לאחר הטעינה ולא נדרש שימוש ב־RUN.

פורמט

RUN [int-num-expr]

SAVE

מיקום במקלדת

S

פקודה/הצהרה

SAVE שולחת תכנית לרשמקול במטרה לאחסנה על סרט.

כיצד להשתמש ב־SAVE

SAVE משמשת בד"כ כפקודה ישירה אך עשויה גם ליצור הצהרה בתכנית. היא מלווה בשם קובץ שהוא משתנה מחרוזת, לדוגמה

SAVE "filename"

שם הקובץ עשוי להכיל עד 10 תוים. בעת הבצוע, ההודעה

Start tape, then press any key

מוצגת על המסך. בלחיצת מקש כלשהו, התכנית נשלחת לרשמקול קלטות, ולבסוף מופיע הדווח ØOK,Ø,1.

שים לב ש־SAVE פועלת בצורה שונה כאשר מחובר כונן זעיר. ראה פרטים במדריך כונן זעיר ומימשק 1.

ריצה אוטומטית

אם יש להריץ אוטומטית את התכנית בזמן הטעינה, יש להשתמש ב־SAVE יחד עם LINE. שם התכנית מלווה ב־LINE וערך מספרי, לדוגמה

SAVE "filename" LINE 1

הערך המלווה את LINE מעוגל לשלם הקרוב ביותר במידת הצורך, ואזי עליו להיות או 1 או מספר שורה בתכנית. התכנית נשלחת לסרט באותה צורה כמו עם SAVE. בעת הטעינה, התכנית רצה באופן אוטומטי מהשורה בעלת המספר המוגדר, או אם לא קיימת שורה כזאת, מהשורה הבאה בתכנית. למעשה, השימוש ב־LINE 1 גורם לכל התכנית לרוץ אוטומטית.

פורמט

SAVE string-expr [LINE int-num-expr]

SAVE CODE

מיקום במקלדת

S
EXTEND MODE
I

פקודה/הצהרה

SAVE CODE שולחת קטע מידע בזכרון לרשמקול קלטות כדי להקליטו על סרט. ניתן תמיד להחזיר את המידע לזכרון בעזרת LOAD CODE.

כיצד להשתמש ב־SAVE CODE

ניתן להשתמש ב־SAVE CODE כפקודה ישירה או ליצירת הצהרה בתכנית. SAVE מלווה בשם קובץ שהוא משתנה מחרוזת ואחריו CODE, המלווה בשני ערכים מספריים המופרדים בפסיק, לדוגמה

SAVE "picture" CODE 16384,6912

שם הקובץ המלווה את SAVE עשוי להכיל עד 1Ø תוים. שני הערכים הבאים אחרי CODE מעוגלים במידת הצורך לשלמים הקרובים ביותר. הערך הראשון נותן את כתובת ההתחלה (16384 לעיל) של המידע בזכרון, והערך השני (6912) נותן את מספר הבתים שיש לאחסן. המידע נשלח לסרט באותה צורה כפי שתכנית נשלחת לסרט ע"י SAVE.

המידע הנשמר בפקודה שלעיל הוא תצוגת המסך.

פורמט

SAVE string-expr CODE
int-num-expr,int-num-expr

SAVE DATA

מיקום במקלדת

S
EXTEND MODE
D

הצהרה/פקודה

SAVE DATA מאחסנת מערך על סרט. ניתן לטעון חזרה את המערך בעזרת LOAD DATA.

כיצד להשתמש ב־SAVE DATA

ניתן להשתמש ב־SAVE DATA ליצירת הצהרה בתכנית או כפקודה ישירה. SAVE מלווה בשם קובץ, אחריו DATA, אות או אות וסימן $, ולבסוף צמד סוגריים ריקים, לדוגמה

45Ø SAVE "numbers" DATA n()
75Ø SAVE "names" DATA n$()

שם קובץ המערך עשוי להכיל עד 1Ø תוים. האות או אות $ שאחרי DATA היא שם המערך בתכנית המיועד לאחסנה על סרט. המערך נשלח לסרט באותה צורה כפי שתכנית נשלחת לסרט ע"י SAVE.

פורמט

SAVE string-expr DATA letter [$]()

SAVE SCREEN$

מיקום במקלדת

S
EXTEND MODE
SYMBOL SHIFT K

פקודה/הצהרה

SAVE SCREEN$ מאחסנת את תצוגת המסך על סרט. ניתן לטעון אותה חזרה למחשב מאוחר יותר בעזרת LOAD SCREEN$.

כיצד להשתמש ב־SAVE SCREEN$

ניתן להשתמש ב־SAVE SCREEN$ כפקודה ישירה או ליצירת הצהרה בתכנית. SAVE מלווה בשם קובץ שהוא משתנה מחרוזת ואחריו SAVE SCREEN$, לדוגמה

SAVE "picture" SCREEN$

שם הקובץ עשוי להכיל עד 1Ø תוים. התצוגה נשלחת לסרט באותו אופן כפי שתכנית נשלחת לסרט ע"י SAVE.

פורמט

SAVE string-expr SCREEN$

SCREEN$

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT K

פונקציה

SCREEN$ מגלה איזה תו מופיע

בנקודה מסוימת על המסך.

כיצד להשתמש ב־SCREEN$

SCREEN$ מלווה בשני ערכים מספריים המופרדים בפסיק ובתוך סוגריים, לדוגמה

```
160 IF SCREEN$ (l,c) = "x"
  THEN PRINT "CRASH"
```

הערכים המלווים את SCREEN$ מעוגלים במידת הצורך לשלם הקרוב ביותר.

הערך הראשון (l שלעיל) עשוי לנוע בין 0 ל־21 ונותן את מספר השורה של נקודה על המסך. הערך השני (c לעיל) עשוי לנוע מ־0 עד 31 ונותן את מספר העמודה של הנקודה. SCREEN$ מחזירה את התו המוצג בנקודה זו כקבוע מחרוזת (התו בתוך סימני מרכאות, "x" לעיל לדוגמה). אם לא קיים כל תו בנקודה זו, SCREEN$ מחזירה מחרוזת אפס (ריקה) ("").

שים לב שניתן גם להשתמש ב־SCREEN$ עם SAVE ו־LOAD לאחסון תצוגת המסך על סרט ולטעון אותה מסרט, ראה SAVE SCREEN$ ו־LOAD SCREEN$ לפרטים.

פורמט

SCREEN$ (int-num-expr,int-num-expr)

SGN סימן

מיקום במקלדת

EXTEND MODE
F

פונקציה

SGN מציינת האם מספר הוא חיובי, שלילי או אפס.

כיצד להשתמש ב־SGN

SGN מלווה בערך מספרי, לדוגמה

```
50 LET x = SGN y
```

בטוי חייב להיות בתוך סוגריים. SGN מחזירה 1 אם ערך הארגומנט (y לעיל) הוא חיובי, -1 אם הוא שלילי ו־0 אם הוא אפס.

פורמט

SGN num-const
SGN num-var
SGN (num-expr)

SIN סינוס

מיקום במקלדת

EXTEND MODE
Q

פונקציה

מה מבצעת SIN?

SIN נותנת את ערך הסינוס של זוית.

כיצד להשתמש ב־SIN

SIN מלווה בערך מספרי, לדוגמה

```
80 LET x = SIN y
```

בטוי חייב להיות בתוך סוגריים. הערך המלווה את SIN הוא הזוית ברדיאנים, ו־SIN מחזירה את סינוס הזוית. ניתן להפוך מעלות לרדיאנים ע"י הכפלה ב־PI/180.

שים לב ש־SIN מחזירה ערך חיובי עבור זויות בין 0 ו־180 מעלות, וערך שלילי עבור זויות בין 180 ו־360 מעלות.

דוגמה

הפקודה

```
PRINT SIN (30 * PI/180)
```

מציגה 0.5, הסינוס של 30 מעלות.

פורמט

SIN num-const
SIN num-var
SIN (num-expr)

SQR שורש רבועי

מיקום במקלדת

EXTEND MODE
H

פונקציה

SQR נותנת את השורש הרבועי של מספר.

כיצד להשתמש ב־SQR

SQR מלווה בערך מספרי, לדוגמה

```
70 LET x = SQR y
```

בטוי חייב להיות בתוך סוגריים. הערך המלווה את SQR (y לעיל) חייב להיות גדול מאפס, ו־SQR מחזירה את השורש הרבועי שלו.

פורמט

SQR num-const
SQR num-var
SQR (num-expr)

STEP

מיקום במקלדת

SYMBOL SHIFT D

ראה FOR

STOP

מיקום במקלדת

SYMBOL SHIFT A

הצהרה/פקודה

STOP עוצרת תכנית בנקודה מסוימת. יתכן ונדרש להשתמש ב־STOP לסיום התכנית הראשית במטרה לבודד את השגרות בקטע נפרד. STOP שימושית מאד גם לצורך ניפוי שגיאות בתכנית.

כיצד להשתמש ב־STOP

STOP משמשת בד"כ ליצירת הצהרה בתכנית. היא מופיעה לחוד, לדוגמה

```
650 STOP
```

בעת הבצוע, התכנית נעצרת והדווח

```
9 STOP statement
```

מופיע עם מספר השורה וההצהרה בה נפסקה התכנית.

נוהלי ניפוי שגיאות, כמו תצוגות ושינוי ערכי משתנים, ניתנים לביצוע לאחר הקשת STOP. הכנסת CONTINUE לאחר מכן, גורמת לתכנית להמשיך בהצהרה הבאה עם הערכים החדשים.

פורמט

STOP

STR$

מיקום במקלדת

EXTEND MODE
Y

פונקציה

STR$ הופכת מספר למחרוזת.

כיצד להשתמש ב־STR$

STR$ מלווה בערך מספרי, לדוגמה

```
90 LET a$ = STR$ x
```

בטוי חייב להיות בתוך סוגריים. STR$ מחזירה את ערך הארגומנט (x לעיל) כקבוע מחרוזת. אם הוקצה ל־x ערך של 65, אזי ההצהרה שלעיל מקצה ל־a$ ערך של "65".

פורמט

STR$ num-const
STR$ num-var
STR$ (num-expr)

TAB

מיקום במקלדת

EXTEND MODE
P

ראה PRINT ,LPRINT

TAN טנגנס

מיקום במקלדת

EXTEND MODE
E

פונקציה

TAN נותנת את ערך הטנגנס של זוית.

כיצד להשתמש ב־TAN

TAN מלווה בערך מספרי, לדוגמה

```
130 LET x = TAN y
```

בטוי חייב להיות בתוך סוגריים. הערך המלווה את TAN הוא ערך הזווית ברדיאנים, ו־TAN מחזירה את טנגנס הזוית. ניתן להפוך מעלות לרדיאנים ע"י הכפלה ב־PI/180.

שים לב ש־TAN מחזירה ערך חיובי עבור זויות בין 0 ו־90 מעלות ובין 180 ו־270 מעלות. עבור זויות בין 90 ו־180 מעלות וזויות בין 270 ו־360 מעלות, TAN מחזירה ערך שלילי.

פורמט

TAN num-const
TAN num-var
TAN (num-expr)

THEN

מיקום במקלדת

SYMBOL SHIFT G

ראה IF

TO

מיקום במקלדת

SYMBOL SHIFT F

פונקציה

ל־TO יש שני שימושים עקריים בבייסיק של סינקלייר. היא משמשת בהקשר עם FOR ליצירת לולאת FOR NEXT (ראה FOR לפרטים) וכמו כן משמשת בפריסת מחרוזות (פיצול מחרוזת לתת־מחרוזות קטנות יותר).

כיצד להשתמש ב־TO עבור פריסת מחרוזות

TO משמשת להגדרת התו הראשון והתו האחרון של תת־מחרוזת בתוך מחרוזת ראשית. לפני TO מופיע ערך מחרוזת, פתיחת סוגריים, ערך מספרי לבחירה חפשית. היא מלווה בערך מספרי אחר אופציונלי, ולבסוף סגירת הסוגריים. לדוגמה

```
80 PRINT a$ (4 TO 7)
```

בטוי מחרוזת חייב להיות בתוך סוגריים. ערך המחרוזת (a$ לעיל) הוא המחרוזת המיועדת לפריסה. שני הערכים המספריים (4 ו־7) מגדירים את מקומות התו הראשון והתו האחרון של תת־המחרוזת בתוך המחרוזת. TO מחזירה את תת־המחרוזת (תוים 4 עד 7 של a$).

לערך המספרי הראשון יש ערך ברירת מחדל של 1 ולשני יש ערך מחדל השווה למיקום התו האחרון במחרוזת. ניתן איפוא להשמיט את הערך הראשון אם תת־המחרוזת מתחילה בתו הראשון במחרוזת, וניתן להשמיט את הערך השני אם תת־המחרוזת מסתיימת בתו האחרון במחרוזת.

פורמט

string-const ([num-expr] TO [num-expr])
string-var ([num-expr] TO [num-expr])
(string-expr) ([num-expr] TO [num-expr])

USR שיגרת משתמש

מיקום במקלדת

EXTEND MODE
L

פונקציה

USR משמשת לקריאת שגרה בקוד מכונה אשר הוכנסה לזכרון בכתובת מסויימת. כמו כן היא משמשת להכנסת נתונים עבור גרפיקה בהגדרת משתמש במקומות השמורים בראש הזכרון.

USR וקוד מכונה

כדי להשתמש בקוד מכונה, USR מלווה בערך מספרי, לדוגמה

```
80 PRINT USR 65000
100 RANDOMIZE USR 65000
```

בטוי חייב להיות בתוך סוגריים. הערך המלווה את USR מעוגל לשלם הקרוב ביותר והופך לכתובת ההתחלה בזכרון שבה הוכנסה שגרת קוד מכונה. כל הצהרה המכילה USR קוראת לשגרה בכתובת זו ו־USR מחזירה את ערך התוכן של צמד אוגר bc.

RANDOMIZE USR או RESTORE USR, לדוגמה, מריצות את השגרה בלבד, בעוד PRINT USR מציגה בנוסף את ערך האוגר bc.

USR וגרפיקה בהגדרת משתמש

כדי ליצור גרפיקה בהגדרת משתמש, משתמשים ב־USR עם POKE. USR מלווה בקבוע או משתנה מחרוזת להחזרת כתובת עבור הצהרת POKE, לדוגמה

```
50 POKE USR "a", 255
```

ערך המחרוזת הבא אחרי USR עשוי להיות אות בודדת בתחום מ־A עד U או a עד u, כאשר אין הבחנה בין אותיות גדולות וקטנות.

USR מחזירה את כתובת ההתחלה של אחד מ־21 חלקי הזכרון השמורים עבור גרפיקה בהגדרת משתמש. כל חלק מכיל 8 כתובות שאליהן מוחדרים בתים ליצירת תו גרפי אחד. ניתן לתת את הבתים בצורה עשרונית או בינארית (ראה BIN).

פורמט

USR int-num-const
USR int-num-var
USR (int-num-expr)
USR string-const
USR string-var

VAL ערך

מיקום במקלדת

EXTEND MODE
J

פונקציה

VAL משנה מחרוזת עם ערך מספרי למספר.

כיצד להשתמש ב־VAL

VAL מלווה בקבוע או משתנה מחרוזת, לדוגמה

```
70 LET x = VAL a$
```

ערך הקבוע או משתנה המחרוזת מופרד מסימני המרכאות, וחייב לכן להיות ערך מספרי. VAL מחשבת אותו, ומחזירה אותו כקבוע מספרי.

דוגמאות

אם ל־a$ יש ערך של "435", אזי ההצהרה שלעיל מקצה ערך של 435 ל־x. ברם, VAL יכולה לחשב גם בטויים, לדוגמה

```
10 INPUT a$ , x
20 PRINT VAL a$
```

ערך המחרוזת המוקצה עבור a$ צריך להיות בטוי המשתמש ב־x, לדוגמה "x * x". ערך מספרי מוקצה עבור x, לדוגמה 5. VAL מסירה את סימני המרכאות לקבלת x * x ומחשבת זאת תוך שימוש בערך שהוקצה עבור x, ומציגה את התוצאה 25.

פורמט

VAL string-const
VAL string-var

VAL$ ערך (מחרוזת)

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT J

פונקציה

VAL$ מחשבת מחרוזת כבטוי מחרוזת.

כיצד להשתמש ב־VAL$

VAL$ מלווה במשתנה מחרוזת, לדוגמה

```
130 PRINT VAL$ a$
```

ערך משתנה המחרוזת מופרד מסימני המרכאות, וחייב לכן להיות בטוי מחרוזת. VAL$ מחשבת את הבטוי ומחזירה את הערך כקבוע מחרוזת.

דוגמאות

נסה תכנית זאת

```
10 INPUT a$ , x$
20 PRINT VAL$ a$
```

משתנה המחרוזת המוקצה עבור a$ צריך להיות בטוי המשתמש ב־x$, לדוגמה "x$ + x$". ערך מחרוזת מוקצה עבור x$, לדוגמה "DO". VAL$ מסירה את המרכאות מערך a$ לקבלת x$ + x$ ומחשבת אותו תוך שימוש בערך שהוקצה ל־x$, ומציגה את התוצאה DODO.

פורמט

VAL$ string-var

VERIFY

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT R

פקודה/הצהרה

VERIFY בודקת האם תכנית אוחסנה כראוי על סרט לאחר בצוע SAVE.

כיצד להשתמש ב־VERIFY

VERIFY משמשת בד"כ כפקודה ישירה בדיוק כמו LOAD ומלווה בשם התכנית, לדוגמה

```
VERIFY "filename"
```

כאשר הסרט מופעל, מוצג על המסך שם כל תכנית הנמצאת בזכרון וכל תכנית על הסרט באותו שם משווה עם התכנית בזכרון. אם שתי התכניות זהות, הדווח

```
0 OK, 0:1
```

ניתן על המסך.

VERIFY מופעלת בצורה שונה כאשר מחובר כונן זעיר. ראה פרטים במדריך כונן זעיר ומימשק 1.

VERIFY CODE ו־VERIFY DATA

ניתן להשתמש ב־VERIFY CODE בדיוק באותו אופן כמו LOAD CODE לאימות אחסון של קטע מידע בזכרון על סרט. VERIFY DATA פועלת באותו אופן כמו LOAD DATA לבדיקת אחסון תקין של מערך על סרט. ראה LOAD CODE ו־LOAD DATA לפרטים נוספים.

פורמט

VERIFY string-expr
VERIFY string-expr CODE
[int-num-expr] [, int-num-expr]
VERIFY string-expr DATA
letter [$] ( )

VERIFY CODE

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT R
EXTEND MODE
I

ראה VERIFY

VERIFY DATA

מיקום במקלדת

EXTEND MODE
SYMBOL SHIFT R
EXTEND MODE
D

ראה VERIFY

דוחות מסך בספקטרום

כאשר הספקטרום עוצר בצוע של בייסיק, מופיע דוח בתחתית המסך. זה מציין שפקודה או תכנית הסתיימו או שאובחנה שגיאה. כל דוח מורכב ממספר או אות קוד המלווה בהודעה קצרה ואחריה מספרי השורה וההצהרה בה נעצר המחשב. פקודה מתוארת כשורה Ø ובתוך שורה, הצהרה 1 היא בתחילת השורה, הצהרה 2 היא אחרי נקודתיים ראשונות או THEN, וכן הלאה. CONTINUE גורמת בד"כ לתכנית להמשיך מההצהרה המוגדרת בדוח.

Ø OK
סיום מוצלח, או נסיון לקפוץ למספר שורה הגדול ממה שקיים בתכנית. CONTINUE מתעלמת מדוח זה וממשיכה בהצהרה המוגדרת בדוח הקודם.

1 NEXT without FOR
קיימת בתכנית מילת המפתח NEXT ללא FOR בהתאמה וקיים משתנה באותו שם כמו משתנה הבקרה.

2 Variable not found
משתנה פשוט היה בשימוש ללא ערך שהוקצה עבורו או ערך שנטען מסרט, או משתנה בקרה שהיה בשימוש עם NEXT ללא קביעתו תחילה בהצהרת FOR, או משתנה מותג שהיה בשימוש לפני הגדרת מימדי המערך בעזרת DIM או טעינת המערך מסרט.

3 Subscript wrong
התג נמצא מחוץ למימדי המערך.

4 Out of memory
אין מספיק מרחב זכרון להשלמת ההצהרה או הפקודה.

5 Out of screen
INPUT יצרה יותר מ־23 שורות לאורך המסך, או שהיה שימוש במספר שורה של 22 ויותר בהצהרת PRINT AT.

6 Number too big
המחשב ניסה ליצור מספר גדול יותר מ־10^{38}.

7 RETURN without GOSUB
מספר הצהרות RETURN גדול ב־1 ממספר הצהרות GOSUB.

8 End of file
דוח טיפול בקבצים בכונן זעיר

9 STOP statement
נעשה שימוש ב־STOP לעצירת התכנית. CONTINUE תגרום להמשך הריצה מההצהרה הבאה.

A Invalid argument
פונקציה קיבלה ארגומנט או ערך שגוי.

B Integer out of range
ערך עוגל לשלם הקרוב ביותר הנמצא מחוץ לתחום המותר.

C Nonsense in BASIC
ההצהרה אינה הגיונית בבייסיק בהקשר של התכנית.

D BREAK-CONT repeats
מקש BREAK נלחץ. CONTINUE תחזור על ההצהרה בה נעצר המחשב.

E Out of DATA
READ ניסתה לקרוא מעבר לסוף הצהרת DATA האחרונה בתכנית.

F Invalid file name
נעשה שימוש ב־SAVE עם שם המכיל יותר מ־1Ø תוים.

G No room for line
לא נותר מרחב זכרון פנוי להכנסת שורת התכנית החדשה.

H STOP in INPUT
STOP הוקשה בתגובה ל־INPUT או בתחילת הנתונים הנקלטים. CONTINUE חוזרת על הצהרת INPUT שנעצרה.

I FOR without NEXT
לולאת FOR NEXT לא בוצעה עקב גבולות או ערך STEP לא נכונים, (לדוגמה, FOR x=5 TO Ø without step) והצהרת NEXT המתאימה לא נמצאה.

J Invalid I/O device
דוח טיפול בקבצים בכונן זעיר.

K Invalid colour
הערך שהוגדר עבור INK, PAPER, FLASH, BRIGHT, INVERSE או OVER או תו הבקרה המתאים מחוץ לתחום המותר.

L BREAK into program
מקש BREAK נלחץ. הדוח מפרט את ההצהרה האחרונה שהיה לבצע וCONTINUE ממשיכה מההצהרה הבאה.

M RAMTOP no good
הערך שהוגדר עבור RAMTOP הוא גדול מדי או קטן מדי.

N Statement lost
המחשב ניסה לקפוץ להצהרה שאינה קיימת עוד.

O Invalid stream
דוח טיפול בקבצים בכונן זעיר.

P FN without DEF
הצהרת FN היתה בשימוש ללא הצהרת DEF FN המתאימה.

Q Parameter error
הצהרת FN מכילה מספר לא נכון של ערכים שיש להעביר לפונקציה, או שאחד הערכים הוא מסוג לא נכון (מחרוזת במקום מספר או להיפך).

R Tape loading error
תהליך הטעינה, מיזוג או אימות נכשל.

מעבר לבייסיק

בייסיק היא שפת מחשב לשימוש כללי אשר פועלת היטב עבור מרבית היישומים. ברם, זוהי אינה שפת המחשב היחידה הניתנת לשימוש בספקטרום. קיימת גם תכנה המאפשרת שפות אחרות כמו FORTH, micro-PROLOG ולוגו. שפות אלה פועלות בצורה שונה לגמרי מבייסיק ופותחות אפשרויות חדשות בפני המחשב.

דווקא בגלל העובדה שבייסיק היא שפה רב־שימושית, היא יכולה להיות מסורבלת למדי ביישומים מסויימים. כמו כן היא די איטית. שפות אחרות יכולות לתת גמישות רבה יותר יחד עם פשטות בתכנות ומהירות ריצה גדולה יותר. FORTH, למשל, מאפשרת לך להגדיר מילים ולהשתמש בהן בהוראות המובנות למחשב אשר יתבצעו פי עשר מהר יותר מאשר בפקודות מקבילות בבייסיק. עם שפת micro-Prolog, המחשב יבין בטויים פשוטים באנגלית. הוא מאחסן אותם בזכרון כבסיס לדיאלוג עם המשתמש. לוגו היא שפת מחשב שפותחה לתחום החינוכי. השפה מציעה פקודות פשוטות ביותר הניתנות לשימוש בצורה גמישה מאד. אך אם ברצונך לכתוב תכניות מהירות באמת עבור הספקטרום +ZX, עליך להבין כיצד לתכנת בקוד מכונה.

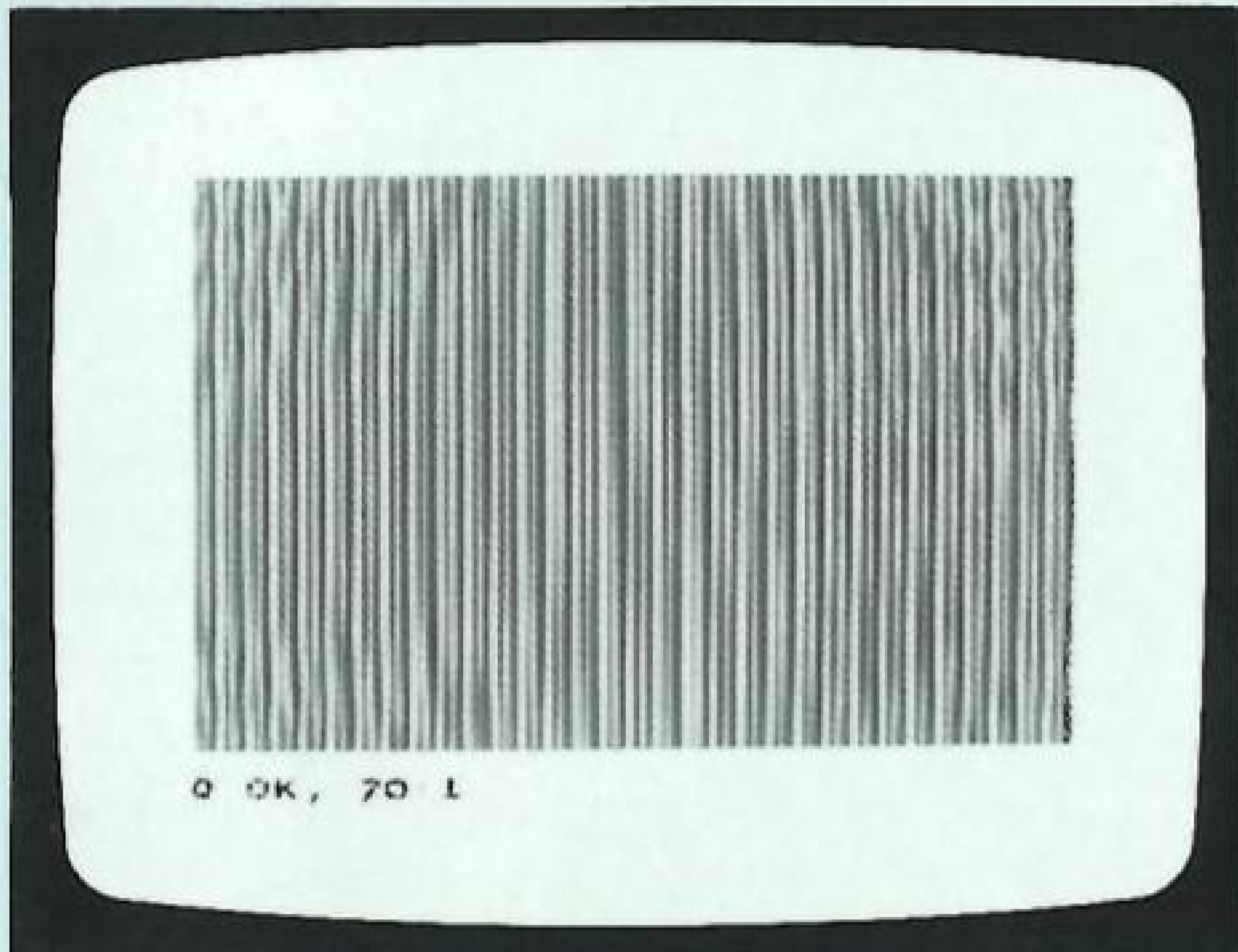
קוד מכונה

שפת בייסיק קיימת על מנת לאפשר לך לתת הוראות למחשב בצורה הנוחה לך להבנה. היע"מ של הספקטרום – שבב Z80A רב העצמה – אינו מבין למעשה בייסיק. חלק בזכרון מכיל תכנית קבועה הנקראת מתורגמן הבייסיק אשר הופכת את הוראותיך בבייסיק לרצף של אותות קוד. קודים אלה מפעילים למעשה את ה־Z80A לביצוע הוראותיך.

למתורגמן נדרש פרק זמן מסויים כדי לתרגם את הוראותיך בבייסיק לקוד Z80A או לקוד מכונה, כפי שהוא נקרא. ברם, אתה יכול לעקוף את מתורגמן הבייסיק ולשלוח ישירות קוד מכונה ל־Z80A. התכנית תתבצע במהירות רבה מאד. המחיר שעליך לשלם תמורת מהירות הבצוע הוא הזמן הנוסף שיש להשקיע בכתיבת תכנית בקוד מכונה. שלא כמו בייסיק, זוהי שפה מאד "לא ידידותית", המחייבת זמן רב ללימוד. תכנות בקוד מכונה הוא מעבר להקף של מדריך זה. ברם, קיימים ספרים רבים המלמדים קוד מכונה בספקטרום ברמה מתקדמת. כדי לקבל מושג על מהירות הביצוע בקוד מכונה, הקש פנימה והרץ את תכנית ההדגמה הבאה.

פסים מיידיים

```
10 FOR x=0 TO 15
20 READ n: POKE 65000+x,n
30 NEXT x
40 DATA 33,255,63,1,1,24,22
50 DATA 55
60 DATA 35,11,120,177,200,114,
24,248
70 RANDOMIZE USR 65000
```



נסה לשנות 55 בשורה 50 לערך כלשהו בין 1 ל־255 וראה כיצד משתנים הפסים. אתה יכול ליצור גם פסים צבעוניים תוך שימוש בפקודת INK בתחילה. אך זוהי אינה מטרת התכנית. ראה כמה מהר נוצרת התצוגה תוך שימוש בקוד מכונה – הבצוע כמעט מיידי. לבייסיק לוקח כמעט שתי שניות למלא את המסך.

תכנית זו פועלת מפני שהצהרת DATA מכילה 16 קודים המאוחסנים בכתובת 65000 והלאה בזכרון בשורות 10 עד 30. שורה 70 שולחת את הקודים ל־Z80A והתצוגה נוצרת מייד. שים לב שהקוד השמיני מבקר את רוחב הפסים.

משחקים רבים שנוצרו עבור הספקטרום כתובים בקוד מכונה ליצירת פעולה מהירה כבזק. כדי לעזור לך לכתוב קוד מכונה, קיימות תכניות הנקראות אסמבלרים. האסמבלר מספק הוראות שאתה מקיש פנימה במקום מספרים בלבד, שאותם בעצם דורש קוד המכונה. ההוראות אינן מילים באנגלית כמו מילות המפתח בבייסיק, אלא קיצורים או כינויים המייצגים את הפעולות שעל המחשב לבצע. עליך, איפוא, להבין כיצד המחשב פועל צעד אחר צעד לפני שתוכל להשתמש בשפת אסף (אסמבלי).

מילון מונחי מחשב

ישנן מילים רבות הנמצאות בשימוש בתחומי המחשב אך מופיעות גם בחיי היומיום במשמעות שונה. להלן מופיעים הסברים לכמה מילים כאלה המופיעות בספר זה, יחד עם מונחי מחשב מיוחדים. למילים המודפסות באותיות מוטות מופיעים פירושים משלהן. אם מופיעה בספר זה מילה או מונח שאינך מבין ואינו נמצא כאן, נסה לחפשו במפתח הממויין (אינדקס).

אוגר – יחידת זכרון קטנה הנפרדת מהזכרון הראשי. האוגרים בתוך ה 'יע"מ' משמשים לביצוע תהליכי החישוב.

אופן פעולה – בספקטרום, זהו אחד מתוך חמישה מצבים, המכתיב אילו 'מילות מפתח' ו 'תוים' ניתן להפיק מכל מקש במקלדת. בזמן התכנות, אופן הפעולה מצויין ע"י אות מהבהבת בתוך ה 'סמן'.

אופרטור – פקודה המבצעת פעולה חשבונית או 'לוגית'.

אמת – כל מצב או תוצאה אשר המחשב מחליט שהם אמת או נכונים. לאמת יש ערך מספרי של 1.

ארגומנט – 'ערך' הנמצא בשימוש 'פונקציה' לקבלת תוצאה מסוימת.

בטוי – צירוף של 'קבועים', 'משתנים' ו 'מילות מפתח'.

בייסיק – שפת המחשב אשר ספקטרום +ZX משתמש בה, וכמוהו גם מרבית המחשבים הביתיים.

בית – מערך של 8 'סיביות' המייצג מספר בעל ערך הנע בין 0 ל־255. כל כתובת ב 'זכרון' מכילה בית אחד.

גלגול – התנועה אשר מאפשרת תצוגה העולה על הגודל המוצג במסך יחיד.

גרפיקה – יצירת דימויים כמו תמונות, טבלאות או דיאגרמות ע"י המחשב.

דוח – הודעה המוצגת ע"י המחשב ומתארת את פעולותיו.

הדפס (PRINT) – להציג תוצאות או 'גרפיקה' על המסך, או להדפיסם במדפסת.

הכנס (ENTER) – לתת הוראה מלאה או מידע למחשב.

הפרדה – דרגת החדות או הפרדת הפרטים האפשרית ב 'גרפיקה' במחשב.

הצהרה – 'מילת מפתח' המשמשת ליצירת פקודה בתוך 'שורה' בתכנית, או הפקודה עצמה.

זכרון – החלק במחשב אשר מכיל את ה 'תכנית' וה 'נתונים' כאשר הם נדרשים, וכמו כן את הוראות ההפעלה הקבועות.

חמרה – המחשב עצמו וכל מכונה או התקן הקשורים בו, כמו 'רכיבים היקפיים'.

טעינה – הזנת 'תכנית' או 'נתונים' למחשב מתוך התקן אחסון כמו קלטת או מחסנית.

יע"מ (CPU) – יחידת עיבוד מרכזית. החלק המרכזי במחשב, אשר מבצע את החישובים ומבקר את היחידות האחרות. בספקטרום +ZX יש מעבד זעיר מסוג Z8Ø.

כתובת – יחידה בודדת של ה 'זכרון'. בספקטרום +ZX קיימות 65536 כתובות.

לוגיקה – תהליך שלפיו המחשב מחליט האם התוצאות הן נכונות או מוטעות או האם מצב הבטוי הנבדק הוא 'אמת' או 'שקר'.

לולאה – קטע ב 'תכנית' החוזר על עצמו פעם אחת או יותר.

מאפיינים – קודים המציינים את צבעי התוים.

מחרוזת – קבוצת "תוים" (אחד או יותר) המובדלת במרכאות כדי לבודדה ממספרים ו 'משתנים מספריים'.

מידע – מילים, מספרים וסימנים בכל צירוף שהוא, אשר המחשב נדרש לטפל בו.

מילת מפתח – הוראה למחשב בשפת בייסיק. יתכן ונדרשים גם מספר 'ערכים' לצורך הפעולה.

מימשק – יחידה המחברת את המחשב ו/או 'רכיבים היקפיים' יחד ומבטיחה אפשרות תקשורת ביניהם.

מערך – קבוצת 'נתונים' הנמצאת יחד באחד מחלקי ה 'זכרון'.

מערך תוים – המערך המושלם של 'תוים' קבועים מראש ומספר קודי בקרה הנמצאים בשמוש המחשב.

משתנה – אחת או יותר יחידות 'זכרון' המייצגות 'קבוע' מסויים לשימוש ע"י המחשב. לכל משתנה ניתן שם או אות לצורך זיהויו. הספקטרום +ZX מבחין בין 'משתנים מספריים' ו 'משתני מחרוזת'.

קוד בינארי – סוג הקוד הנמצא בשימוש המחשב. הקוד כולל סדרת מצבי "גע" - "תק" – לדוגמה, דחפים חשמליים במצבי "גע" - "תק".

קוד מכונה – השפה אותה מבין הספקטרום +ZX. תכניות בייסיק מתורגמות לקוד מכונה בזמן הרצתן ע"י המחשב.

קינון – ארגון 'לולאות' בתוך תכנית כך שלולאה אחת או יותר מתבצעות בתוך לולאה אחרת.

קלט – 'תכניות' ו'נתונים' המוזנים למחשב.

ראם (RAM) – זכרון בגישה אקראית. חלק ב'זכרון' היכול להכיל 'תכנית' ו'נתונים' וערכים משתנים אחרים. ידוע גם כזכרון נדיף (לא קבוע). תוכן זכרון ראם נמחק כאשר מתנתקת אספקת הזרם. לספקטרום +ZX יש 58 ק' ראם.

רום (ROM) – זכרון קריאה בלבד. החלק ב'זכרון' אשר מכיל 'תכניות' קבועות ופקודות למחשב. לספקטרום +ZX יש 16 ק' רום.

רכיב היקפי – כל רכיב המחובר למחשב.

רשום – 'שורות' בתכנית המפורטות לפי הסדר ברשימה על המסך או מדפסת.

שורה – פקודה או מערך פקודות ב'תכנית'. לכל שורה ניתן מספר כך שהיא תתבצע בנקודה המדוייקת בתוך סדר פעולת כל השורות.

שמור (SAVE) – אגירת 'תכנית' או 'נתונים' בהתקן אחסון כמו קלטת או תקליטון.

שקר – כל מצב או תוצאה אשר המחשב מחליט שהוא לא אמת או לא נכונה. לשקר יש ערך מספרי של 0.

שרשור – צירוף מספר 'מחרוזות' יחד ע"י הוספתן אחת לאחת.

תו – אות בודדת כלשהי, מספר (0 עד 9), סימן או יחידת 'גרפיקה' הניתנים לתצוגה או הדפסה.

תחביר – הסדר הנכון של 'מילות מפתח', 'קבועים', 'משתנים', ו'בטויים' הדרושים לצורך יצירת פקודת בייסיק תקפה.

תכנה – כל 'תכנית', כולל תכניות קבועות בזכרון 'רום' או מחסניות.

תכנית – קבוצת פקודות בסדר מסויים המתבצעת ע"י המחשב.

משתנה מחרוזת – משתנה אשר מייצג 'מחרוזת'. משתני מחרוזת בנויים תמיד מאות אחת וסימן דולר ($).

משתנה מספרי – משתנה אשר מייצג מספר. משתנים מספריים מורכבים מאות אחת או יותר.

נתונים – 'מידע' המגיע למחשב מתוך 'תכנית' או המוזן מבחוץ לתוך המחשב לצורך קבלת תוצאות.

סיבית – מצב "גע" או "תק" ב'קוד בינארי'. זהו שם קיצור לספרה בינארית.

סמן – המקום על פני המסך בו עתיד להיות מוצג התו הבא. ניתן לסמנו ע"י סימן מהבהב המציין את 'אופן הפעולה' בו נמצא המחשב.

עורך (EDIT) – שינוי פרטים בתוך 'תכנית'.

ערך – כל מספר או מחרוזת אשר ניתן להציב בהם או לייצגם ע"י 'קבוע', 'משתנה' או 'בטוי'.

פונקציה – תהליך שבו המחשב לוקח אחד או יותר 'ערכים' (או 'ארגומנטים') ומשתמש בהם לקבלת תוצאה שהיא ערך אחר.

פיקסל – נקודת הצבע הקטנה ביותר המופיעה על המסך. זהו שם מקוצר (מאנגלית) ל'תא תמונה'.

פלט – תוצאות המופקות ע"י המחשב.

פקודה – הוראה בודדת המתבצעת ע"י המחשב.

פקודה ישירה – מערך של הוראה אחת או יותר המתבצע מיידית עם הגיעו למחשב.

ק' – מידת קיבול ה'זכרון' של מחשב. 1 ק' שווה ל־1 קילרביית או 1024 סיביות. קיבול הזכרון ב־ק' שווה למספר הכללי של 'כתובות' בזכרון, כשכל אחת יכולה לאחסן סיבית אחת. לספקטרום +ZX יש 48 ק' 'ראם' ו־16 ק' 'רום', הנותנים בסה"כ 64 ק'.

קבוע – מספר או קבוצה של אות אחת או יותר, או כל 'תו' אחר.

מפתח ממוין (עברית)

מספרי עמודים מודגשים מתייחסים לאיורים ומסגרות.

פורסם לראשונה ב־1984 ע"י דורלינג קינדרסליי בע"מ, רח' הנריאטה 9, לונדון, בשותפות עם סינקלייר ריסרץ' בע"מ, דרך ויליס 25, קמברידג'.

עורך דוד ברני
עורך אמנותי פטר לף
תכנון דברה לי
צילום טרוור מלטון
צלם צילומי רשת וינסנט אוליבר
ניהול עריכה אלן בקינגהם
סדר צילום טכנוסדר בע"מ

תורגם ע"י חבר המתרגמים, תל־אביב

רשתות נקודות תצוגת מסך

רשת הנקודות שלהלן מראה קואורדינטות עבור גרפיקה בהפרדה גבוהה וגרפיקה בהפרדה נמוכה. למרות שמילות מפתח גרפיות מסויימות פועלות עם קואורדינטות בהפרדה גבוהה או נמוכה בלבד, אתה יכול להשתמש במילות מפתח עבור הפרדה גבוהה ונמוכה יחד בעת יצירת תצוגה.

רשת הנקודות בהפרדה נמוכה מחולקת לשני חלקים, שטח התצוגה הראשי ושני קווים בתחתית המסך. PRINT AT תיצור תוים בחלק העליון ו־INPUT AT בחלק התחתון. הקואורדינטות עבור מסך בהפרדה נמוכה רשומות בצד העליון והצד השמאלי של רשת הנקודות.

רשת הנקודות בהפרדה גבוהה תופסת רק את שטח התצוגה הראשי. PLOT, DRAW ו־CIRCLE משמשות ליצירת גרפיקה בהפרדה גבוהה. הקואורדינטות עבור הפרדה גבוהה רשומות בצד התחתון והצד הימני של רשת הנקודות.

עמודות

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

שורות: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

קואורדינטות אנכיות: 175 168 160 152 144 136 128 120 112 104 96 88 80 72 64 56 48 40 32 24 16 8 0

0 8 16 24 32 40 48 56 64 72 80 88 96 104 112 120 128 136 144 152 160 168 176 184 192 200 208 216 224 232 240 248 255

קואורדינטות אפקיות

רשת נקודות לגרפיקה בהגדרת משתמש

הוסף את המספרים הנתונים לקבלת הערך העשרוני עבור כל טור ברשת, והעזר ב־POKE USR לתכנות התו הרצוי.

128 64 32 16 8 4 2 1

סה"כ טור

תוכנת ספקטרום

כל מגוון התוכנה עבור מחשבי ספקטרום (כולל כל הנושאים הקיימים כיום) תואם בשלמותו למחשב החדש ספקטרום +ZX.

דורלינג קינדרסלי בע"מ
בשיתוף עם
סינקלייר ריסרץ' בע"מ